NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

ANALYSIS OF
A DISTRIBUTED DECISION ALGORITHM

by

Sung Chu Hahn

December 1985

Thesis Advisor: C. W. Therrien

Approved for public release; distribution is unlimited

ITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

PORT SECURITY CLASSIFICATION NCLASSIFIED	1b. RESTRICTIVE MARKINGS
CURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
ECLASSIFICATION/DOWNGRADING SCHEDULE	
FORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
AME OF PERFORMING ORGANIZATION al Postgraduate School; 6b. OFFICE SYMBOL (If applicable) 62	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
DDRESS (City, State, and ZIP Code) terey, California 93943-5100	7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5100
AME OF FUNDING/SPONSORING RGANIZATION; 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
DDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO; PROJECT NO; TASK NO; WORK UNIT ACCESSION NO

TLE (Include Security Classification)
LYSIS OF A DISTRIBUTION DECISION ALGORITHM

ERSONAL AUTHOR(S)
Hahn, Sung Chu

TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
ster's Thesis	FROM ______ TO ______	1985 December	78

UPPLEMENTARY NOTATION

COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Centralized Decision Algorith; Distributed Decision Algorithm A; Distributed Decision Algorithm B

BSTRACT (Continue on reverse if necessary and identify by block number)

Distributed decision problems arise whenever two or more sensors d their associated computers must work cooperatively to make a cision about a commonly observed event. Typical examples are in rget detection and classification. The problem is usually aracterized by a limited bandwidth of the communication link tween the sensors.

This thesis develops and evaluates an algorithm for distributed cision and compares it to a non-distributed or centralized form of e algorithm. Analysis of the algorithm is carried out for some

DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
NAME OF RESPONSIBLE INDIVIDUAL ofessor C. Therrien	22b TELEPHONE (Include Area Code) 408 646-2160; 22c. OFFICE SYMBOL 62Ti

FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete — SECURITY CLASSIFICATION OF THIS PAGE

low-dimensional cases. Computer simulations were carried out for higher dimensional cases. The simulation work was done in Fortran under CMS on an IBM 370/3033 computer.

Analysis of a Distributed Decision Algorithm

by

Sung Chu Hahn
Major, R.O.K. Air Force
B.S., R.O.K. Air Force Academy, 1976

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1985

ABSTRACT

Distributed decision problems arise whenever two or more sensors and their associated computers must work cooperatively to make a decision about a commonly observed event. Typical examples are in target detection and classification. The problem is usually characterized by a limited bandwidth of the communication link between the sensors.

This thesis develops and evaluates an algorithm for distributed decision and compares it to a non-distributed or centralized form of the algorithm. Analysis of the algorithm is carried out for some low-dimensional cases. Computer simulations were carried out for higher dimensional cases. The simulation work was done in Fortran under CMS on an IBM 370/3033 computer.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. GENERAL DISCUSSION

This thesis presents an algorithm for distributed decision and compares its performance to that of a centralized decision rule. A distributed decision rule is characterized by the fact that a decision algorithm is distributed between processors of two or more sensors.

For simulation and evaluation, some programs were written in Fortran on an IBM 770/3033 computer. The work of this thesis is concerned with the analysis of the distributed decision rule only. A related thesis by Capt. Mark Schon [Ref. 1] is concerned with the implementation in real time on a distributed microcomputer system.

The specific goals of this thesis are to :

1 Develop and analyze a specific distributed decision algorithm.

2 Generate all necessary data, parameters and statistics to simulate the decision algorithms.

3 Experimentally evaluate the capabilities and performance of a distributed decision rule and compare it with a centralized decision rule.

B. BACKGROUND

In this thesis statistical methods are used to develop decision algorithms. Since we deal with many observations which represent data collected by the sensors, vector notation and matrix algebra is used extensively in these algorithms.

The Gaussian distribution is used to characterize the observations because this provides a decision rule that is relatively easy to analyze and develop intuition. It also provides a reasonable decision rule based on second moment statistics (mean and covariance) of the observation data.

Bayes's rule is used to develop decision algorithms for binary decision (class 1 or class 2) and to develop the decision boundary concept. Mathematical manipulation of Bayes's rule leads to specific decision algorithms which are analyzed and evaluated in the computer simulation.

Since it is very difficult to visualize decision boundaries in high dimensional spaces, we have developed some computer programs to experimentally evaluate the algorithms. The simulations show that in many cases the distributed decision algorithms are quite reliable and perform nearly as well as a centralized decision algorithm.

C. STRUCTURE OF THE THESIS

The remainder of this thesis is structured as follows. Chapter II addresses the overall processes of the decision rule including probability laws for random vectors and Bayes decision theory. The matrix algebra needed to describe this is also developed. Decision rules are interpreted as providing boundaries and regions in a multidimensional space that determine decisions made about the observed data.

Chapter III describes a distributed decision algorithm and the form of its decision boundary. Detailed analysis and evaluation are given comparing it with the centralized decision rule.

Chapter IV presents computer simulations to test the distributed decision rules. To simulate data collected by sensors, an autoregressive time series model is introduced. Second moment statistics i.e. the mean, variance and covariance of the given random vectors are computed by a statistical estimation algorithm. These statistics are further used to compute the algorithm parameters. Decision algorithms are tested with the generated data and results are given.

Chapter V summarizes the results of the thesis and describes the capabilities and performance of the decision algorithms. Suggestions are also given for future research.

II. BASIC DECISION PROCESSES

A. CLASS DECISION

Class decision means a classification of objects into categories. The objects of interest may be radar targets, electronic waveforms or signals, printed letters or characters, states of a system, or any number of other things that are desired to be classified.

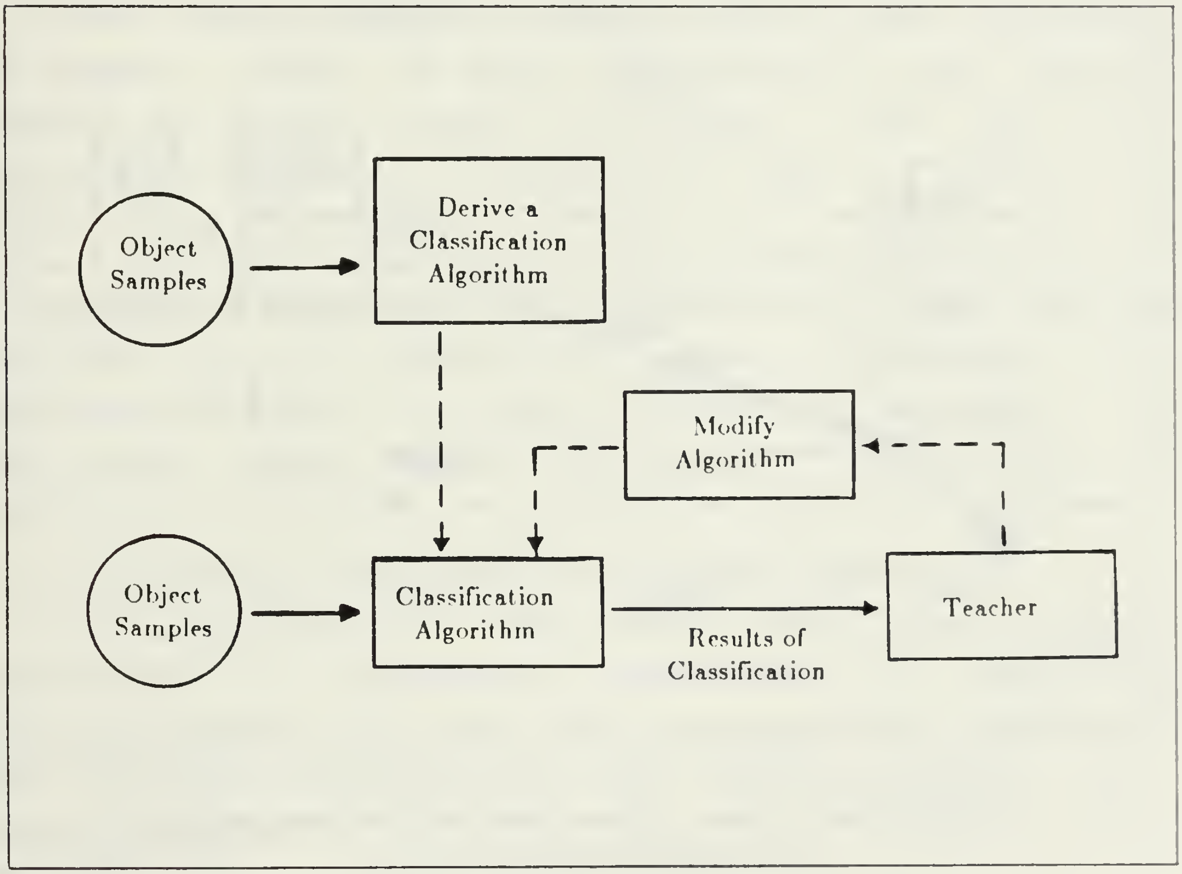

Figure 2.1 Basic Class Decision Procedure

In testing a class decision algorithm the individual classes of objects are presumed already known. The basic procedure for a class decision is illustrated in Fig. 2.1. A portion of a known set of labeled objects is extracted and

used to derive a classification algorithm. These objects comprise the "training set".

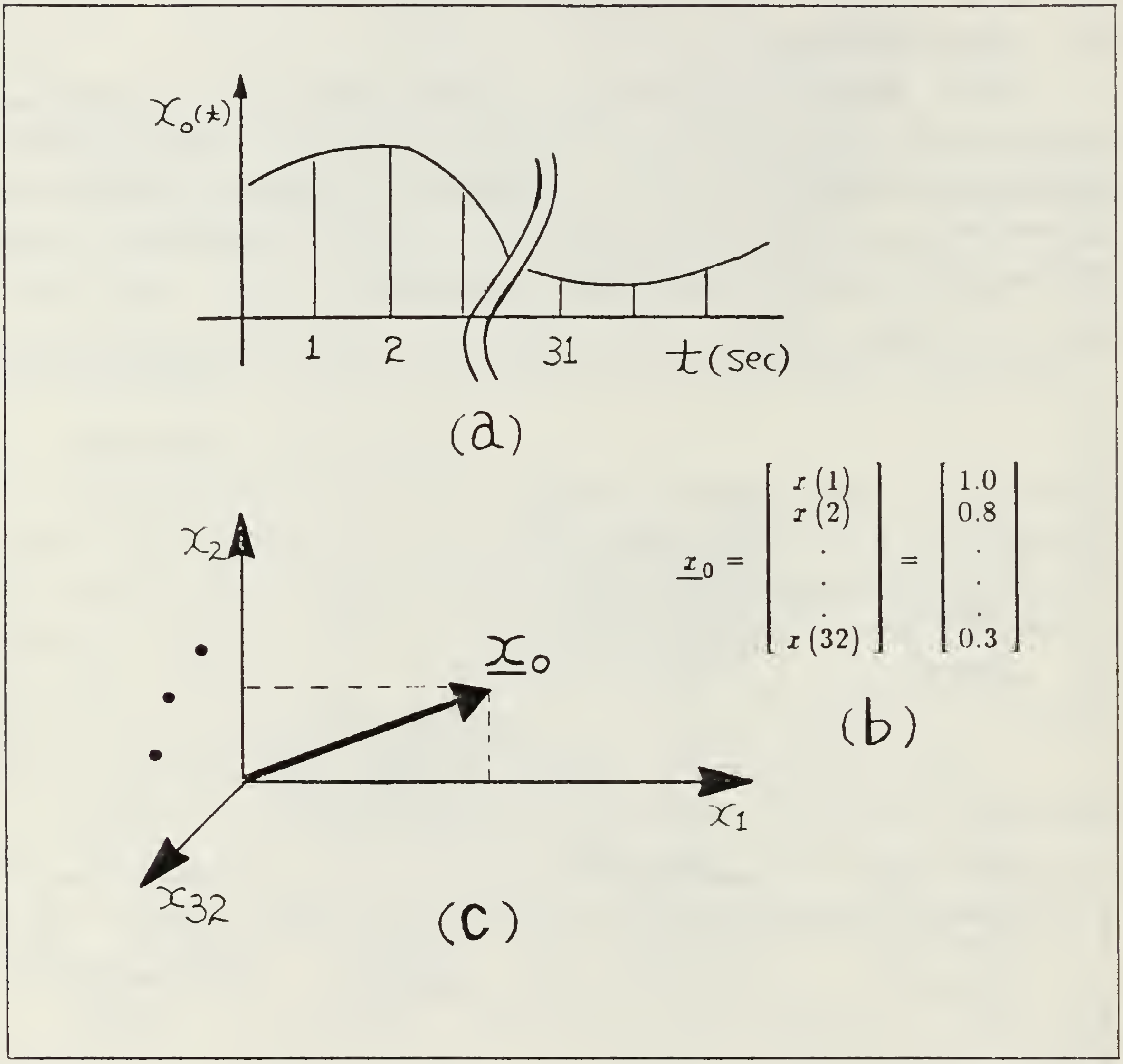

Figure 2.2 (a) A Waveform to be Recognized
(b) Observation Vector
(c) Depiction of Observation Space

The remaining objects are then used to test the classification algorithm and these are collectively referred to as the "test set". The performance of the algorithm can be evaluated because the correct classes of the individual

objects in the test set are known. The result of classification is supervised by a teacher who may dictate suitable modifications to the algorithm.

A simple example of a class decision is presented to illustrate its approach and to define some relevant concepts. Fig. 2.2(a) illustrates 32-dimensional observations of electronic waveforms. The vector $\underline{x}_o$ is called the observation vector and the multidimensional space in which it resides is called the observation space. These are depicted in Fig. 2.2(b) and (c).

Every problem in class decision has at least two things in common. First, an exact description of the various classes of objects cannot be obtained. Thus the class decision is inherently a probabilistic topic. Secondly, the objects are represented by vectors in a multidimensional space. Thus the observation vectors of the objects to be classified are multidimensional random vectors which must be described in a statistical sense. Similarly, the performance of the algorithm must also be measured in a statistical sense. Thus an adequate background in probability and statistics is important for these problems.

B. THE GAUSSIAN DISTRIBUTION FOR RANDOM VECTORS

In engineering and many other areas, the Gaussian distribution is frequently encountered. It describes certain phenomena well with just two parameters, namely the mean and the covariance of the random variables. The Gaussian density function for one-dimensional random variables is:

$$p_x(x) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left[-\frac{1}{2} \frac{(x - m_x)^2}{\sigma_x^2} \right] \tag{2.1}$$

Fig. 2.3 shows a one-dimensional density function $p_x(x)$ with its mean value m_x and variance $\sigma_x{}^2$.

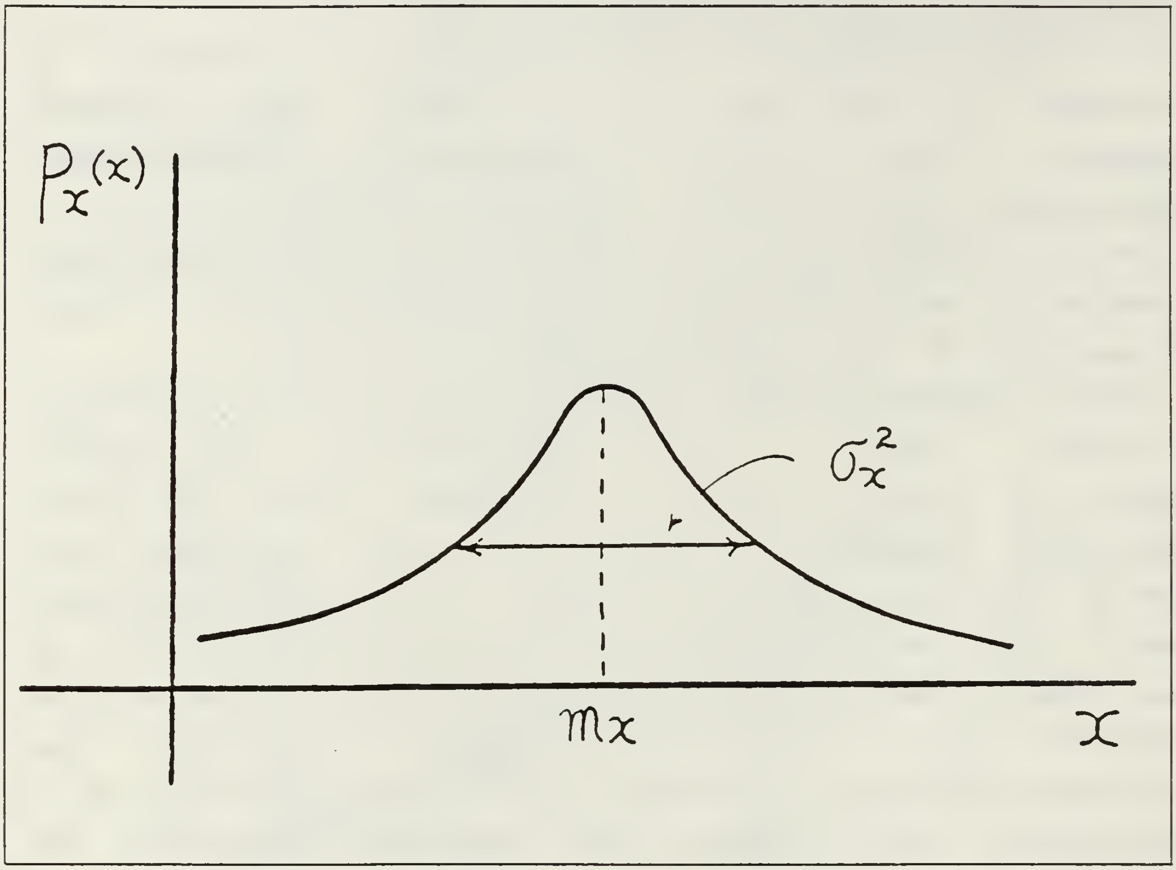

Figure 2.3 One Dimensional Gaussian Density Function

In the two-dimensional case (i.e. two random variables) the Gaussian density function is:

$$p_{x,y}(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp\left[\frac{1}{2(1-\rho^2)} \left\{ \frac{(x-m_x)^2}{\sigma_x^2} + 2\rho\frac{(x-m_x)(y-m_y)}{\sigma_x\sigma_y} + \frac{(y-m_y)^2}{\sigma_y^2} \right\} \right] \qquad (2.2)$$

Fig. 2.4 shows a two-dimensional density function $p_{x,y}(x,y)$ with its mean values m_x and m_y, its variances σ_x^2 and σ_y^2 and the correlation coefficient ρ of both random variables x and y [Ref. 2: p. 158].

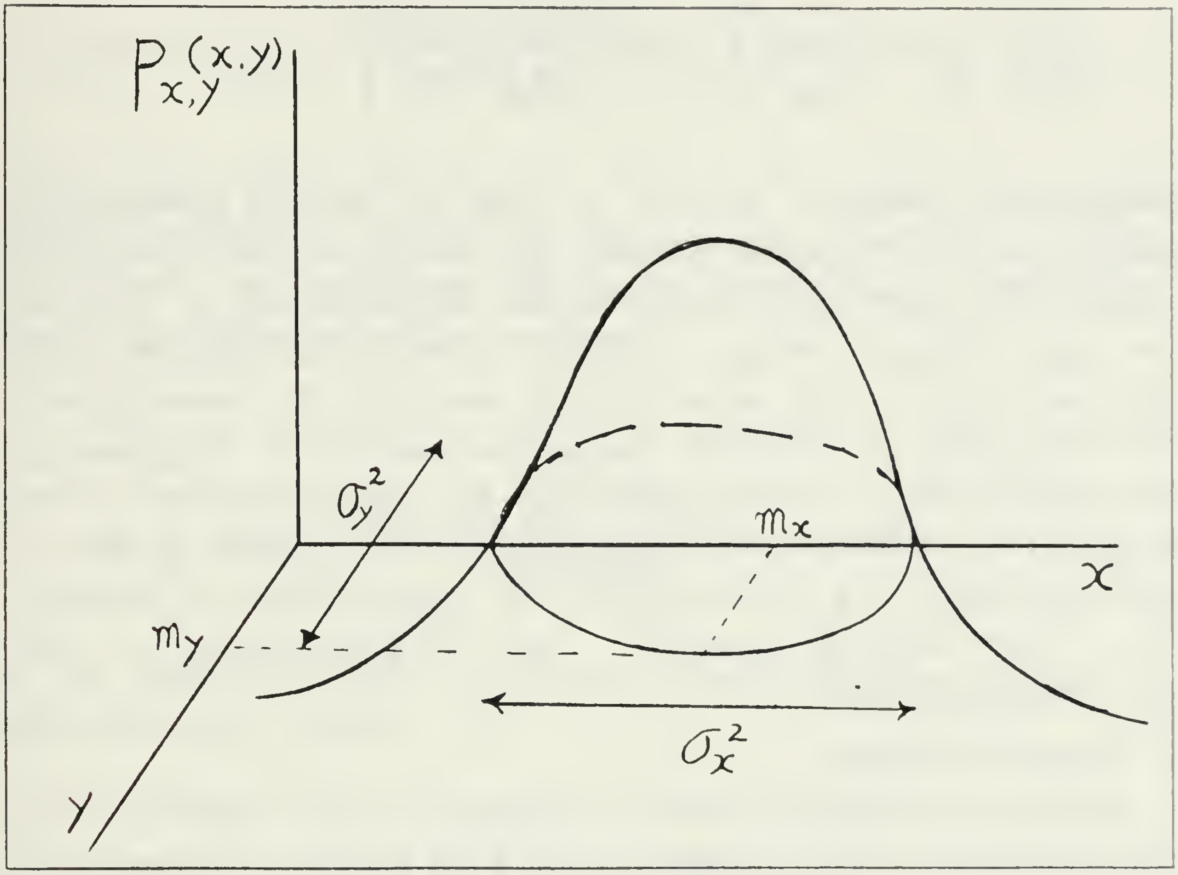

Figure 2.4 Two-Dimensional Gaussian Density Function

The Gaussian density function for two sets of multidimensional random variables $\underline{x}$ and $\underline{y}$ is expressed by the combined vector $\underline{z}$ and its parameters as follows:

$$p_{\underline{z}}(\underline{z}) = \frac{1}{(2\pi)^{\frac{N}{2}} \, | \mathrm{K} |^{\frac{1}{2}}} \exp\left[-\frac{1}{2} (\underline{z} - \underline{m})^T \mathrm{K}^{-1} (\underline{z} - \underline{m}) \right] \qquad (2.3)$$

where

$$\underline{z} = \begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix}, \quad \underline{m}^{(i)} = \begin{bmatrix} \underline{m}_x^{(i)} \\ \underline{m}_y^{(i)} \end{bmatrix}, \quad \mathbf{K}^{(i)} = \begin{bmatrix} \mathbf{K}_x^{(i)} & \mathbf{B}_{xy}^{(i)} \\ \mathbf{B}_{xy}^{(i)T} & \mathbf{K}_y^{(i)} \end{bmatrix}, \quad i = 1,2 \qquad (2.4)$$

Observation vectors $\underline{x}$ and $\underline{y}$ are N and M-dimensional respectively. The mean vectors $\underline{m}_x$ and $\underline{m}_y$ are also N and M dimensional, and they represent the expectations of vectors i.e. $\underline{m}_x = E[(\underline{x})]$ and $\underline{m}_y = E[(\underline{y})]$. The covariance matrices $[K_x]$ and $[K_y]$ are of size [N X N] and [M x M] respectively and represent correlations among the components of $\underline{x}$ and $\underline{y}$. The matrix $[B_{xy}]$ is of size [N x M] and represents cross correlation between the components of the vectors $\underline{x}$ and $\underline{y}$. These matrices are also defined by expectations of vectors i.e. $K_x = E[(\underline{x}-\underline{m}_x)(\underline{x}-\underline{m}_x)^T]$, $K_y = E[(\underline{y}-\underline{m}_y)(\underline{y}-\underline{m}_y)^T]$, and $B_{xy} = E[(\underline{x}-\underline{m}_x)(\underline{y}-\underline{m}_y)^T]$.

C. BAYES'S THEOREM

Bayes's theorem is used to convert prior probabilities into posterior probabilities. The form of this theorem that is useful to us is:

$$p_r(\omega \mid \underline{x}) = \frac{p(\underline{x} \mid \omega) p_r(\omega)}{p(\underline{x})} \qquad (2.5)$$

where ω represents an event such as "object belongs to class 1". The term $p_r(\omega)$ is called the prior probability of the event and the term $p_r(\omega \mid \underline{x})$ is called the posterior probability. More generally, let ω_1, ω_2, ... , ω_n be n mutually exclusive classes exhausting the set of all

possible classes of the objects. Then the conditional probability law gives this following equation:

$$p_r(\omega_i \mid \underline{x}) = \frac{p(\underline{x} \mid \omega_i)p_r(\omega_i)}{p(\underline{x})}, \quad i = 1,2, \ldots, n \qquad (2.6)$$

where $p(x) = \sum_{i=1}^{n} p(\underline{x}|\omega_i)P_r(\omega_i)$. If we consider the case where observations consist of two vectors $\underline{x}$ and $\underline{y}$ and assume that there are only two classes, class $1(\omega_1)$ and class $2(\omega_2)$, the above equation becomes:

$$p_r(\omega_i \mid \underline{x},\underline{y}) = \frac{p_{\underline{x},\underline{y}|\omega_i}(\underline{x},\underline{y} \mid \omega_i)p_r(\omega_i)}{p_{\underline{x},\underline{y}}(\underline{x},\underline{y})}, \quad i = 1,2 \qquad (2.7)$$

If we make a class decision based on the posterior probabilities, that is

$$p_r(\omega_1 \mid \underline{x},\underline{y}) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} p_r(\omega_2 \mid \underline{x},\underline{y}) \qquad (2.8)$$

then Eqs. 2.7 and 2.8 lead to the likelihood ratio test

$$l(\underline{x},\underline{y}) = \frac{p_1(\underline{x},\underline{y})}{p_2(\underline{x},\underline{y})} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{p_r(\omega_2)}{p_r(\omega_1)} = T \qquad (2.9)$$

where we have used the notation $p_i(\underline{x},\underline{y})$ to represent the class conditional density $p(\underline{x},\underline{y}|\omega_i)$. If the likelihood

ratio $l(\underline{x},\underline{y})$ for specific observation vectors $\underline{x}$ and $\underline{y}$ is greater than a threshold value T then class $1(\omega_1)$ is chosen. On the other hand if the ratio is less than T class $2(\omega_2)$ is chosen.

D. DECISION BOUNDARY OF CENTRALIZED DECISION RULE

Although any decision rule for our problem is at least two-dimensional, corresponding to observations x and y, it is still instructive to look at the likelihood ratio for a single variable x. The decision boundary of a one-dimensional case is relatively simple as Fig. 2.5 shows.

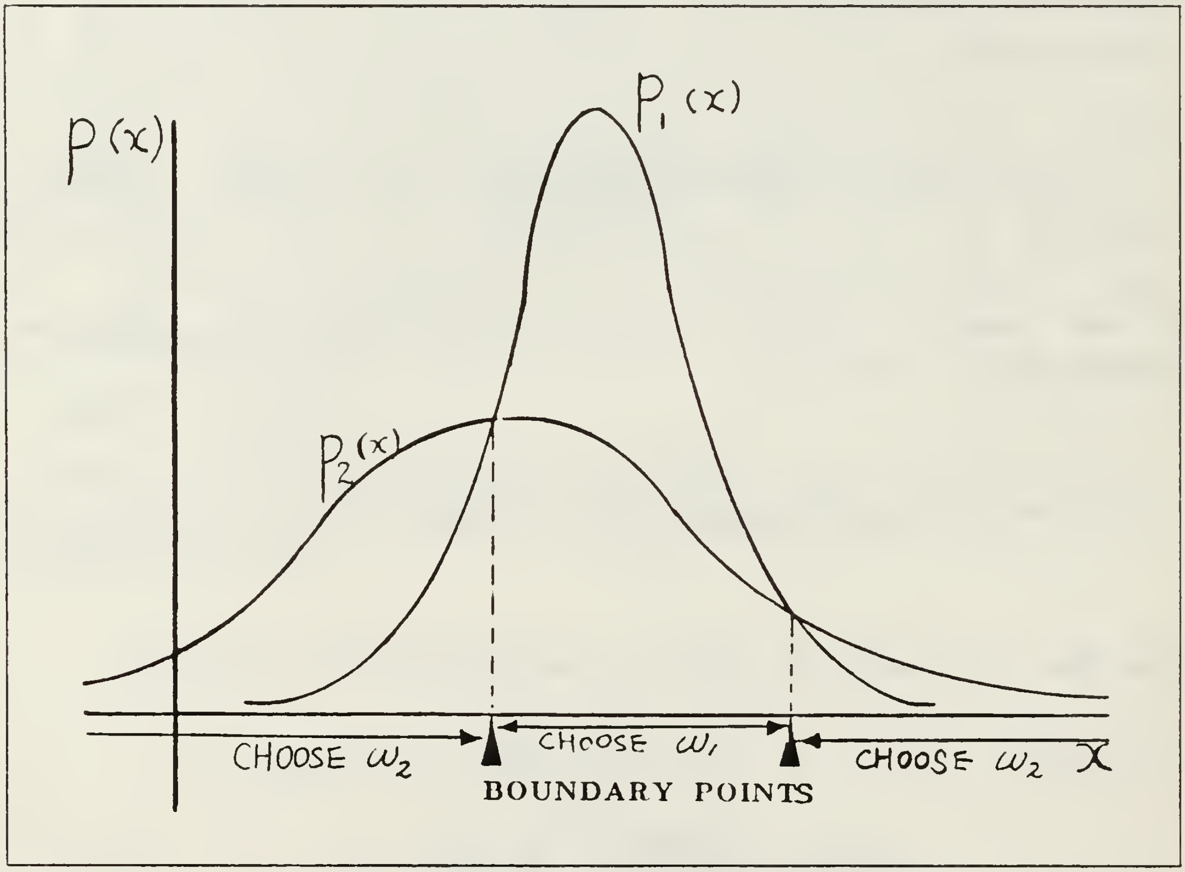

Figure 2.5 Decision Boundary of One-Dimensional Case

The decision boundary is just given as a set of points on the x axis.

In the two-dimensional case the decision boundary is more complicated. For Gaussian random vectors it could be a straight line, ellipse, hyperbola, parabola or a combination.

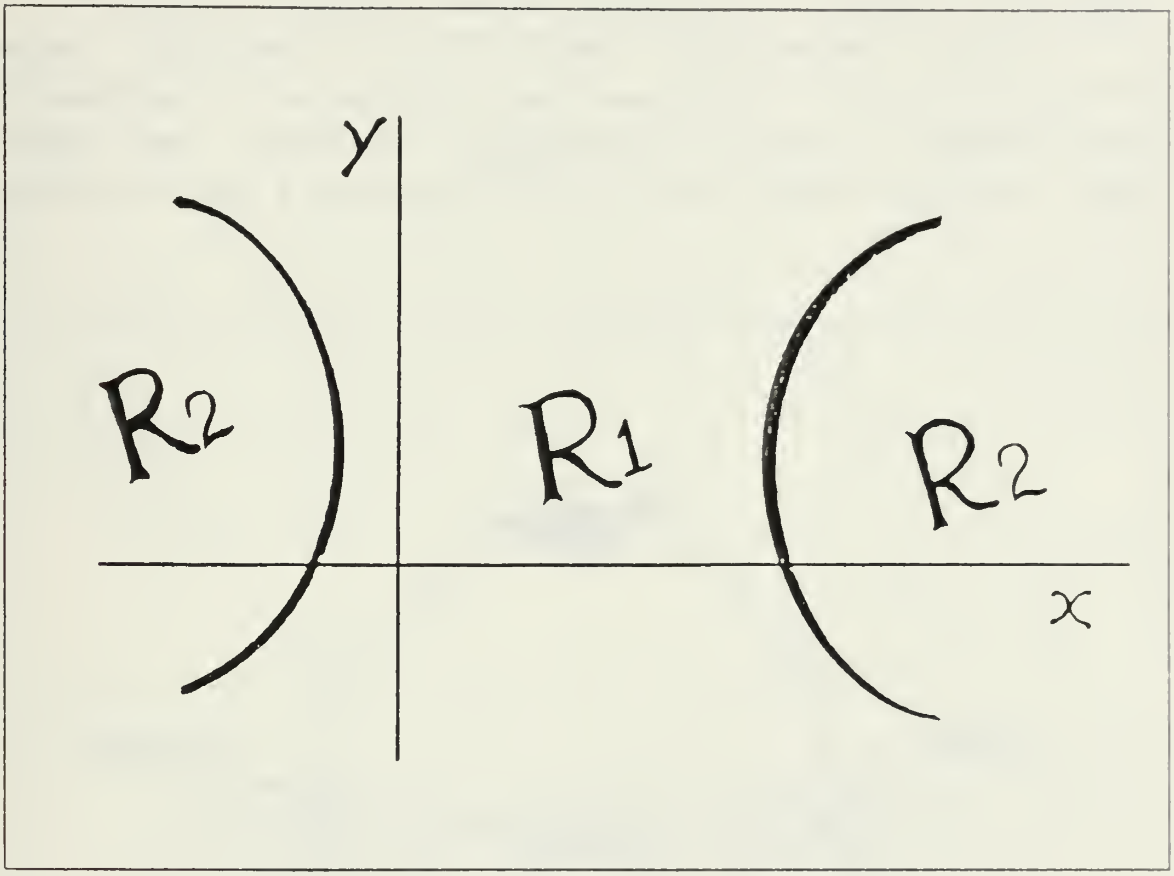

Figure 2.6 Decision Boundary of Two-Dimensional Case (Hyperbola)

Fig. 2.6 shows an example, if observation variables x and y are outside the curve lines i.e. in region $1(R_1)$ the decision is class 1, if inside i.e. in region $2(R_2)$ the decision is class 2.

When the dimension of the observations is more than two, it is more difficult to visualize the decision boundary but the concept is still useful. A centralized decision rule uses the $\underline{x}$ and $\underline{y}$ vectors together directly in its algorithm.

All equations use joint probability densities such as $p_1(\underline{x},\underline{y})$, $p_2(\underline{x},\underline{y})$ which determine the multidimensional decision boundary.

III. DISTRIBUTED DECISION RULE

A. BACKGROUND

The AEGIS weapons system simulation project, currently being conducted at the Naval Postgraduate School, is attempting to determine the feasibility of replacing the larger and relatively expensive mainframe computer, the AN/UYK-7, with a system of 16 or 32 bit VLSI computers [Ref. 3].

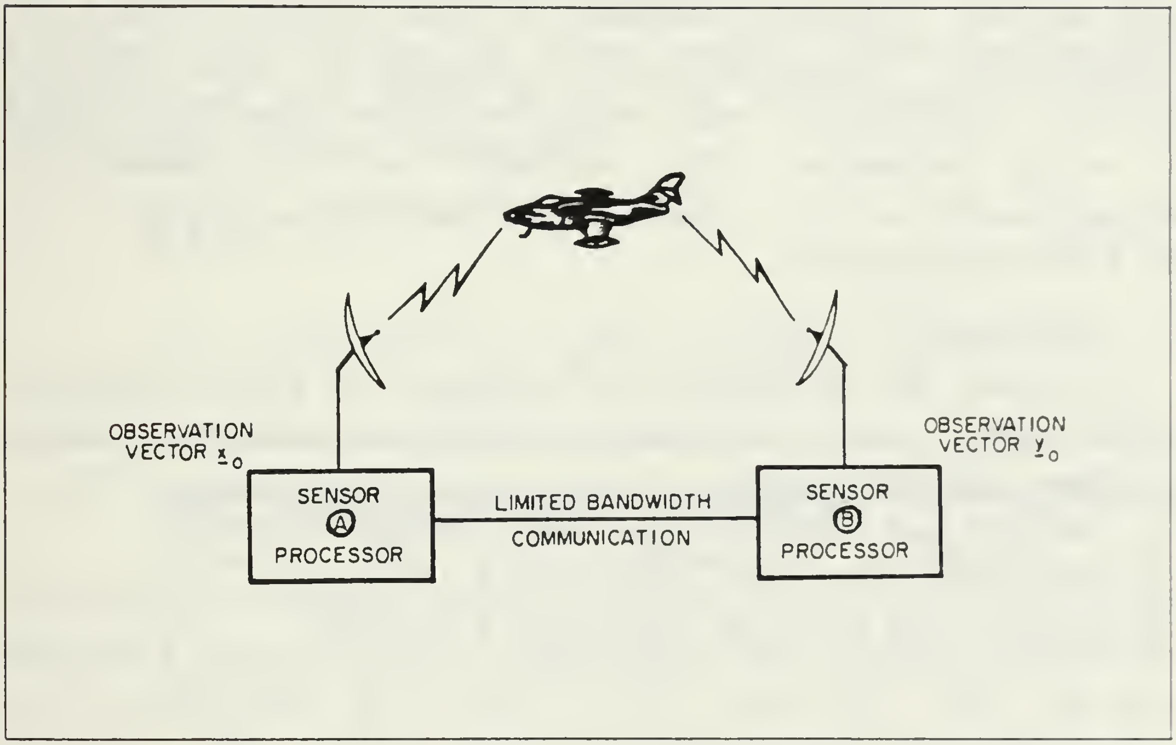

Figure 3.1 Distributed Decision Scenario

As the capabilities and performance of microcomputers continue to improve, it is becoming apparent that an integrated multiprocessor system of less expensive, compact microcomputers can manage many real-time applications that have previously used mainframe computers. This set of

microcomputers has been used to demonstrate our distributed decision rule in a realistic environment[Ref. 1]. The computers have been organized to simulate two sensors observing the same object for purposes of detection and/or classification.

As illustrated in Fig. 3.1, sensor A deals with the observation vector $\underline{x}_o$ only, while sensor B deals with the observation vector $\underline{y}_o$ exclusively. A centralized decision rule uses both observation vectors $\underline{x}_o$ and $\underline{y}_o$ at once in a single processor to determine its decision. In a distributed decision procedure, each processor cannot use both vectors together because of the limited bandwidth communication. Nevertheless, by exchange of some minimum essential information, each processor makes a decision which is quite reliable. The concepts will be developed mathematically in this chapter and tested experimentally in the following chapter.

B. DEFINITION

In order to introduce the concepts of three decision algorithms here each algorithm is presented mathematically.

These algorithms are:

1 Centralized Decision Algorithm (C.D.A)

2 Distributed Decision Algorithm A (D.D.A)

3 Distributed Decision Algorithm B (D.D.B)

1. Centralized Decision Algorithm

The concept of a likelihood ratio was introduced in Chapter 2 Section C. From the likelihood ratio the centralized decision rule is derived. The likelihood ratio for Gaussian data is expressed (using Eq. 2.3 and Eq. 2.9) as follows:

$$l(\underline{z}) = \frac{p_1(\underline{z})}{p_2(\underline{z})} =$$

$$\frac{|\mathbf{K}^{(1)}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\underline{z} - \underline{m}^{(1)})^T K^{(1)^{-1}}(\underline{z} - \underline{m}^{(1)})\right]}{|\mathbf{K}^{(2)}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\underline{z} - \underline{m}^{(2)})^T K^{(2)^{-1}}(\underline{z} - \underline{m}^{(2)})\right]} \qquad (3.1)$$

$$\underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{p_r(\omega_2)}{p_r(\omega_1)} = T$$

where vector $\underline{z}$, $\underline{m}^{(i)}$, and matrix $[K^{(i)}]$ were introduced in Eq. 2.4. Here the subscript 1 and 2 means class 1 and class 2 respectively in the two class case. Taking the natural logarithm of both sides of Eq. 3.1 yields this following centralized decision algorithm:

$$\frac{1}{2}\Big[(\underline{z} - \underline{m}^{(2)})^T K^{(2)^{-1}}(\underline{z} - \underline{m}^{(2)}) \qquad (3.2)$$

$$- (\underline{z} - \underline{m}^{(1)})^T K^{(1)^{-1}}(\underline{z} - \underline{m}^{(1)}) + \ln \frac{|\mathbf{K}^{(2)}|}{|\mathbf{K}^{(1)}|}\Big] \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \ln T$$

Such a centralized decision procedure is shown in Fig. 3.2.

2. <u>Separation of Centralized Decision Algorithm into $\underline{x}_o$ and $\underline{y}_o$ Observation Vector Components</u>

Although Eq. 3.2 adequately represents the centralized decision rule, we want to put it in a form involving vectors $\underline{x}_o$, $\underline{y}_o$ separately and certain partitions of the matrices $K^{(1)}$, $K^{(2)}$, $\underline{m}^{(1)}$, and $\underline{m}^{(2)}$ for the two classes. This will help us to develop the distributed decision rules and enable us to more directly compare the distributed rules to the centralized rule. Fig. 3.2 shows a scenario using both observation vectors in a centralized processor. To develop a distributed form of the decision algorithm, we proceed as follows. Using a conditional probability law the joint probability $p(\underline{x},\underline{y})$ is equivalent to:

$$p_i(\underline{x},\underline{y}) = p_i(\underline{x})\, p_i(\underline{y}\,|\,\underline{x}), \quad i=1,2 \tag{3.3}$$

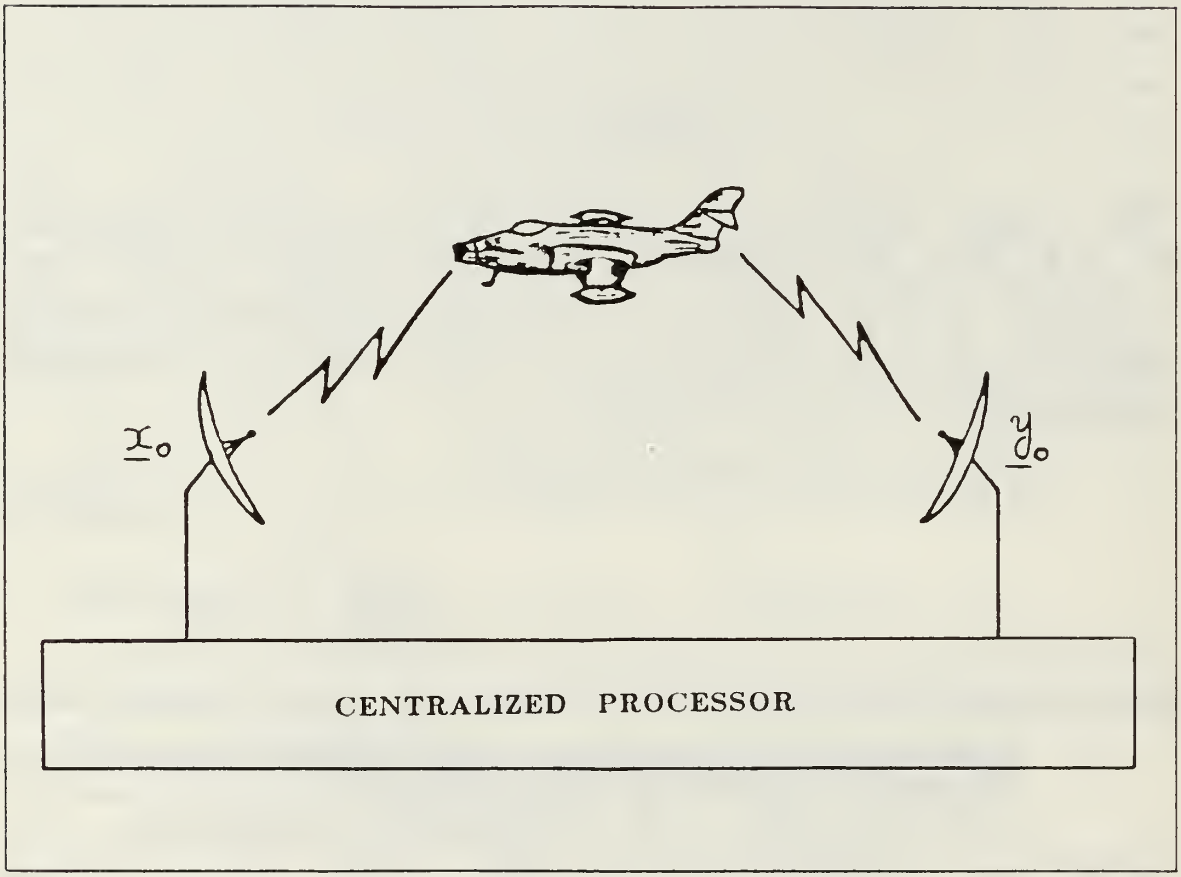

Figure 3.2 Centralized Decision Scenario

Taking the log base e of both sides leads to:

$$\ln\ p_i(\underline{x},\underline{y}) = \ln\ p_i(\underline{x}) + \ln\ p_i(\underline{y}\,|\,\underline{x}), \quad i=1,2 \tag{3.4}$$

Eq. 3.4 shows how the probability density can be distributed into two parts, where one part is a function of $\underline{x}$ only and the other part is a function of $\underline{y}$ given $\underline{x}$. For the Gaussian case the probability density function of random vector $\underline{x}$ is:

$$p_i(\underline{x}) = \frac{1}{(2\pi)^{\frac{N}{2}} \, |\mathbf{K}_x^{(i)}|^{\frac{1}{2}}} \tag{3.5}$$

$$\exp\left[-\frac{1}{2} [\underline{x} - \underline{m}_x^{(i)}]^T \, [K_x^{(i)}]^{-1} \, [\underline{x} - \underline{m}_x^{(i)}] \right], \quad i = 1,2$$

The conditional probability density function of vector $\underline{y}$ given $\underline{x}$[Ref. 2] is:

$$p_i(\underline{y} \mid \underline{x}) = \frac{1}{(2\pi)^{\frac{N}{2}} \, |\mathbf{K}_{y|x}^{(i)}|^{\frac{1}{2}}} \tag{3.6}$$

$$\exp\left[-\frac{1}{2} [\underline{y} - \underline{m}_{y|x}^{(i)}]^T \, [K_{y|x}^{(i)}]^{-1} \, [\underline{y} - \underline{m}_{y|x}^{(i)}] \right], \quad i = 1,2$$

where

$$\mathbf{K}_{y|x}^{(i)} = \mathbf{K}_y^{(i)} - \mathbf{B}_{xy}^{(i)T} \, [\mathbf{K}_x^{(i)}]^{-1} \, \mathbf{B}_{xy}^{(i)}, \quad i = 1,2 \tag{3.7}$$

and

$$\underline{m}_{y|x}^{(i)} = \underline{m}_y^{(i)} + [\mathbf{B}_{xy}^{(i)}]^T [\mathbf{K}_x^{(i)}]^{-1}[\underline{x} - \underline{m}_x^{(i)}] \ , \quad i=1,2 \qquad (3.8)$$

In Eqs. 3.7 and 3.8, $[K_{y|x}]$ and $\underline{m}_{y|x}$ is easily calculated using all parameters and both observation vectors $\underline{y}_o$ and $\underline{x}_o$ directly. Thus the conditional probability density function $p(\underline{y}|\underline{x})$ is determined without any difficulties. Using the above expressions Eqs. 3.5 and 3.6, Eq. 3.1 becomes:

$$\frac{p_1(\underline{x},\underline{y})}{p_2(\underline{x},\underline{y})} = \frac{p_1(\underline{x})\, p_1(\underline{y}\,|\,\underline{x})}{p_2(\underline{x})\, p_2(\underline{y}\,|\,\underline{x})}$$

$$= \frac{|\,\mathbf{K}_x^{(1)}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\,[\underline{x} - \underline{m}_x^{(1)}]^T\ [\mathbf{K}_x^{(1)}]^{-1}\,[\underline{x} - \underline{m}_x^{(1)}]\right]}{|\,\mathbf{K}_x^{(2)}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\,[\underline{x} - \underline{m}_x^{(2)}]^T\ [\mathbf{K}_x^{(2)}]^{-1}\,[\underline{x} - \underline{m}_x^{(2)}]\right]}$$

$$\frac{|\,K_{y|x}^{(1)}\,|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\,[\underline{y} - \underline{m}_{y|x}^{(1)}]^T\ [\mathbf{K}_{y|x}^{(1)}]^{-1}\,[\underline{y} - \underline{m}_{y|x}^{(1)}]\right]}{|\,K_{y|x}^{(2)}\,|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\,[\underline{y} - \underline{m}_{y|x}^{(2)}]^T\ [\mathbf{K}_{y|x}^{(2)}]^{-1}\,[\underline{y} - \underline{m}_{y|x}^{(2)}]\right]} \qquad (3.9)$$

$$\underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{p_r(\omega_2)}{p_r(\omega_1)} = T$$

Finally by taking the natural logarithm of both sides, Eq. 3.9 becomes:

$$\lambda_A(\underline{x}_0) + \lambda_B(\underline{y}_0 | \underline{x}_0) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \ln T \qquad (3.10)$$

where

$$\lambda_A(\underline{x}_0) = \frac{1}{2}\Bigg[[\underline{x}_0 - \underline{m}_x^{(2)}]^T [\mathbf{K}_x^{(2)}]^{-1} [\underline{x}_0 - \underline{m}_x^{(2)}] - [\underline{x}_0 - \underline{m}_x^{(1)}]^T [\mathbf{K}_x^{(1)}]^{-1} [\underline{x}_0 - \underline{m}_x^{(1)}] + \ln \frac{|\mathbf{K}_x^{(2)}|}{|\mathbf{K}_x^{(1)}|} \Bigg] \qquad (3.11)$$

$$\lambda_B(\underline{y}_0 | \underline{x}_0) = \frac{1}{2}\Bigg[[\underline{y}_0 - \underline{m}_{y|x}^{(2)}]^T [\mathbf{K}_{y|x}^{(2)}]^{-1} [\underline{y}_0 - \underline{m}_{y|x}^{(2)}] - [\underline{y}_0 - \underline{m}_{y|x}^{(1)}]^T [\mathbf{K}_{y|x}^{(1)}]^{-1} [\underline{y}_0 - \underline{m}_{y|x}^{(1)}] + \ln \frac{|\mathbf{K}_{y|x}^{(2)}|}{|\mathbf{K}_{y|x}^{(1)}|} \Bigg] \qquad (3.12)$$

Eq. 3.10 suggests a distributed form for the decision rule which is described in the next section.

3. Distributed Decision Rule A

Fig. 3.1 shows that processor A uses vector $\underline{x}_o$ only and processor B uses vector $\underline{y}_o$ only. In this distributed decision rule the processor A which is to compute $\lambda_A(\underline{x}_o)$ has no problem because it observes vector $\underline{x}_o$ directly and it

has all the other parameters needed in Eq. 3.11. Processor B, which is to compute $\lambda_B(\underline{y}_o|\underline{x}_o)$, has a problem however because it does not have direct access to $\underline{x}_o$. This other observation vector appears in Eq. 3.8; thus Eq. 3.12 is dependent on $\underline{x}_o$.

If there exists a way to estimate the observation vector $\underline{x}_o$ using known parameters and sensor B's own observation vector $\underline{y}_o$, then the estimated $\underline{x}$ which we denote by $\hat{\underline{x}}_i$ can be used in Eq. 3.8 instead of $\underline{x}_o$ itself. This procedure is known as a generalized likelihood ratio test [Ref. 4]. In this case sensor B will have no problem in the computation since it is assumed that the other parameters necessary to compute $\underline{m}_{y|x}$ and $K_{y|x}$ are already known.

To obtain an estimate $\hat{\underline{x}}_i$, processor B considers the following conditional density:

$$p_i(\underline{x} \mid \underline{y}) = \frac{1}{(2\pi)^{\frac{N}{2}} \mid \mathbf{K}_{x|y}^{(i)} \mid^{\frac{1}{2}}} \exp\left[-\frac{1}{2} [\underline{x} - \underline{m}_{x|y}^{(i)}]^T [\mathbf{K}_{x|y}^{(i)}]^{-1} [\underline{x} - \underline{m}_{x|y}^{(i)}] \right], \quad i=1,2 \tag{3.13}$$

In particular processor B chooses $\underline{x}$ as the value that maximizes $p_i(\underline{x}|\underline{y})$. Because of its Gaussian form, Eq. 3.13 is maximized when $\underline{x} = \underline{m}_{x|y}$. From the symmetry of Eqs. 3.6 and 3.13 the following estimate is obtained(see Eq. 3.8).

$$\hat{\underline{x}}_i = \underline{m}_{x|y}^{(i)} = \underline{m}_x^{(i)} + \mathbf{B}_{xy}^{(i)} [\mathbf{K}_y^{(i)}]^{-1} [\underline{y}_o - \underline{m}_y^{(i)}], \quad i=1,2 \tag{3.14}$$

Now processor B can use $\underline{x}$ which is calculated by known parameters $\underline{m}_x$, $[B_{xy}]$, $[K_y]$, $\underline{m}_y$, and its own observation vector $\underline{y}_o$ in Eq. 3.10 to implement a distributed decision algorithm. In this algorithm Eq. 3.10 is modified to the form:

$$\lambda_A(\underline{x}_0) + \lambda_B'(\underline{y}_0) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \ln T \tag{3.15}$$

where

$$\lambda_B'(\underline{y}_0) = \lambda_B(\underline{y}_0 | \underline{\hat{x}}_i) \tag{3.16}$$

and where $\lambda_B(\underline{y}_o|\underline{\hat{x}}_i)$ is given by Eq. 3.12 with $\underline{x}_o$ replaced by $\underline{\hat{x}}_i$ of Eq. 3.14. Specifically $\underline{\hat{x}}_1$ will be used in the computation of $\underline{m}^{(1)}{}_{y|x}$ and $\underline{\hat{x}}_2$ will be used in the computation of $\underline{m}^{(2)}{}_{y|x}$ as these terms appear in Eq. 3.12. The term $\lambda_A(\underline{x}_o)$ is exactly the same as in Eqs. 3.10 and 3.11.

Let us summarize the the results as follows. In this distributed decision rule A $\lambda_A(\underline{x}_o)$ is the same as was shown in the centralized decision rule of Eq. 3.10. However $\lambda'_B(\underline{y}_o)$ is different from $\lambda_B(\underline{y}_o|\underline{x}_o)$ in the centralized decision rule. Actually $\lambda'_B(\underline{y}_o)$ is simplified notation for the term $\lambda_B(\underline{y}_o|\underline{\hat{x}}_i)$. Both $\lambda_A(\underline{x}_o)$ and $\lambda'_B(\underline{y}_o)$ are single statistics which must be added together and compared to the threshold value T to decide the class of the observed object. These statistics $\lambda_A(\underline{x}_o)$ and $\lambda'_B(\underline{y}_o)$ are displayed in Eq. 3.11 and 3.16.

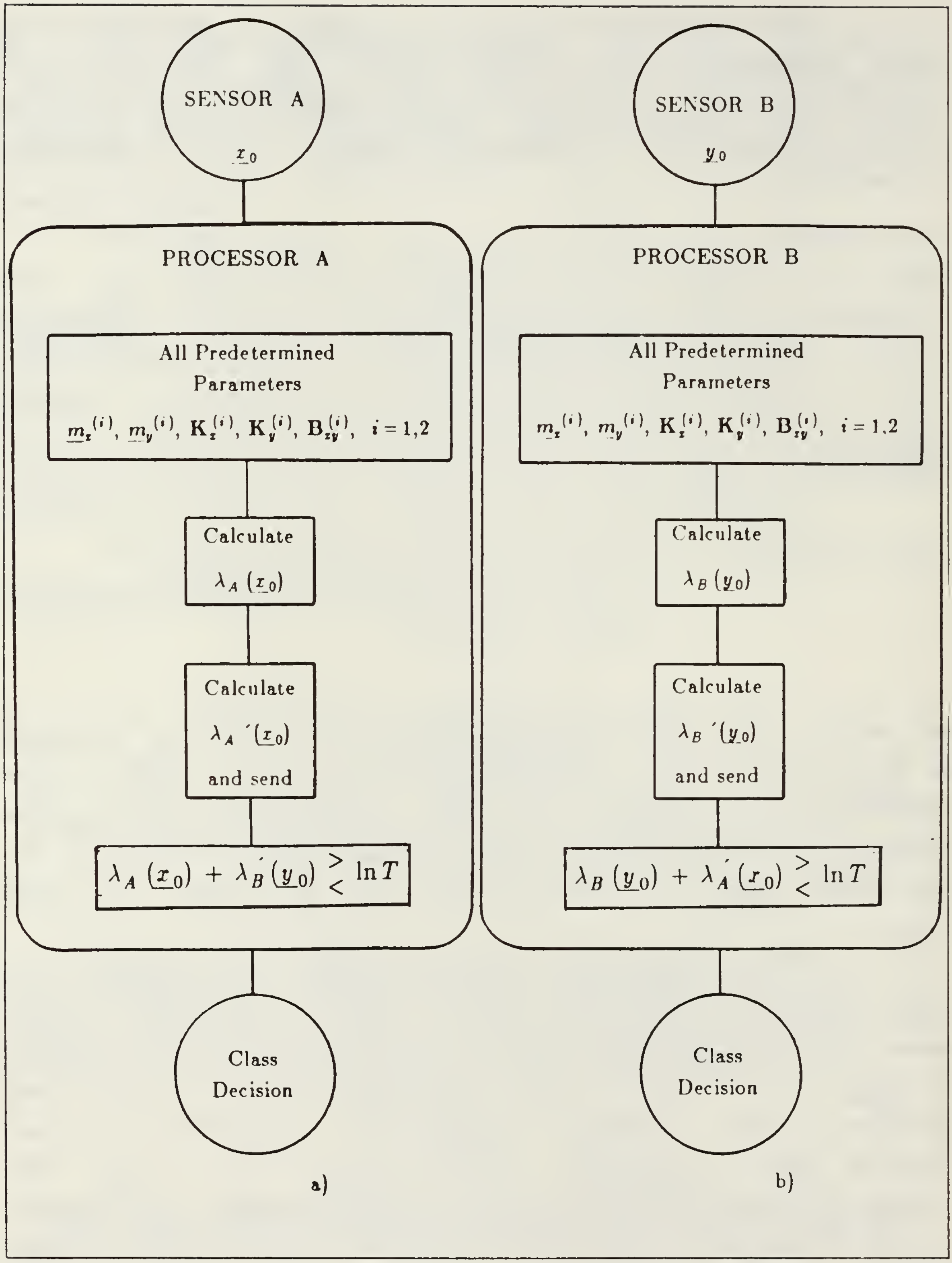

Figure 3.3 Block Diagram of Distributed Decision Algorithms (a) Type A(D.D.A) (b) Type B(D.D.B)

The single statistic $\lambda'_B(\underline{y}_o)$ which is calculated in processor B is transmitted to processor A through the limited bandwidth communication link. Processor A will then have both its own calculated statistic $\lambda_A(\underline{x}_o)$ and the statistic $\lambda'_B(\underline{y}_o)$ received from processor B. Therefore it can decide the class of observed object using Eq. 3.15. Eq. 3.15 is called distributed decision rule A because the class decision is made in processor A. This algorithm is illustrated in Fig. 3.3 (a).

4. <u>Distributed Decision Rule B</u>

Distributed decision rule A was considered in the previous section. A symmetric form of this algorithm is illustrated in Fig. 3.3(b). This algorithm uses a symmetric form of the conditional probability law of Eq. 3.3.

$$p_i(\underline{x},\underline{y}) = p_i(\underline{y})\, p_i(\underline{x} \mid \underline{y}), \quad i=1,2 \qquad (3.17)$$

which leads to:

$$\ln\, p_i(\underline{x},\underline{y}) = \ln\, p_i(\underline{y}) + \ln\, p_i(\underline{x} \mid \underline{y}), \quad i=1,2 \qquad (3.18)$$

By analogy and symmetry with the equations used in distributed decision algorithm A, the following algorithm is derived

$$\lambda_B(\underline{y}_0) + \lambda_A'(\underline{x}_0) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \ln T \qquad (3.19)$$

where

$$\lambda_B(\underline{y}_0) = \frac{1}{2}\Bigg[[\underline{y}_0 - \underline{m}_y^{(2)}]^T [\mathbf{K}_y^{(2)}]^{-1} [\underline{y}_0 - \underline{m}_y^{(2)}] - [\underline{y}_0 - \underline{m}_y^{(1)}]^T [\mathbf{K}_y^{(1)}]^{-1} [\underline{y}_0 - \underline{m}_y^{(1)}] + \ln \frac{|\mathbf{K}_y^{(2)}|}{|\mathbf{K}_y^{(1)}|} \Bigg] \tag{3.20}$$

$$\lambda_A'(\underline{x}_0) = \frac{1}{2}\Bigg[[\underline{x}_0 - \underline{m}_{x|\hat{y}_2}^{(2)}]^T [\mathbf{K}_{x|y}^{(2)}]^{-1} [\underline{x}_0 - \underline{m}_{x|\hat{y}_2}^{(2)}] - [\underline{x}_0 - \underline{m}_{x|\hat{y}_1}^{(1)}]^T [\mathbf{K}_{x|y}^{(1)}]^{-1} [\underline{x}_0 - \underline{m}_{x|\hat{y}_1}^{(1)}] + \ln \frac{|\mathbf{K}_{x|y}^{(2)}|}{|\mathbf{K}_{x|y}^{(1)}|} \Bigg] \tag{3.21}$$

where $K_{x|y}$ and $\underline{m}_{x|y}$ are computed from equations analogous to Eqs. 3.7 and 3.8. Processor B calculates the single statistic $\lambda_B(\underline{y}_o)$ using its own observation vector $\underline{y}_o$. Processor A computes the single statistic $\lambda'_A(\underline{x}_o)$ using the following estimate for the vector $\underline{y}$:

$$\underline{\hat{y}}_i = \underline{m}_{y|x}^{(i)} = \underline{m}_y^{(i)} + \mathbf{B}_{xy}^{(i)T} [\mathbf{K}_x^{(i)}]^{-1} [\underline{x} - \underline{m}_x^{(i)}] \ , \quad i = 1,2 \tag{3.22}$$

Thus $\lambda'_A(\underline{x}_o)$ is a simplified notation for $\lambda_A(\underline{x}_o|\hat{\underline{y}}_i)$ and is transmitted to processor B through the communication link. Therefore processor B computes $\lambda_B(\underline{y}_o)$ locally and receives $\lambda'_A(\underline{x}_o)$ from processor A. Then processor B makes a decision about the class of the observed object using Eq. 3.19. This procedure represents distributed decision rule B because the class decision is made by processor B.

C. COMPARISON WITH THE CENTRALIZED DECISION RULE

Three algorithms were introduced and explained in the previous sections A and B. Table 1 shows the differences among them very briefly. Notice that the two forms (Type A and Type B) given for the centralized decision rule are equivalent. In distributed decision algorithm A, processor B uses the estimated value $\hat{\underline{x}}_i$ instead of the observed value $\underline{x}_o$ and sends the result $\lambda'_B(\underline{y}_o)$ to processor A. In distributed decision algorithm B, processor A uses $\hat{\underline{y}}_i$ instead of $\underline{y}_o$ and sends $\lambda'_A(\underline{x}_o)$ to B. These differences are visualized simply in Table 2.

Use of the estimates $\hat{\underline{x}}_i$ in distributed decision algorithm A, and $\hat{\underline{y}}_i$ in distributed decision algorithm B makes the results of these algorithms different from each other and different from the centralized decision rule. Further, the use of rules A and B together can result in an ambiguous situation where the two decisions are different. This can be resolved in a number of ways discussed later.

The key components which make the algorithms different from one another are the use of the estimate $\hat{\underline{x}}_i$ in distributed decision algorithm A, and $\hat{\underline{y}}_i$ in distributed decision algorithm B. If the estimated vectors $\hat{\underline{x}}_i$ and $\hat{\underline{y}}_i$ are close to the actual observation vectors $\underline{x}_o$ and $\underline{y}_o$ respectively then the results of the distributed algorithms A and B would be close to each other and close to the centralized algorithm. Although we have not been able to characterize theoretically the relative performance of these

algorithms we can show their results experimentally on a number of different test cases. These results are given in the next chapter.

TABLE 1

DIFFERENCES AMONG THREE ALGORITHMS

DECISION ALGORITHM	MATHEMATICAL FORM	DESCRIPTION & COMMENTS
1. CENTRALIZED DECISION ALGORITHM	. TYPE A $\lambda_A(\underline{x}_0) + \lambda_B(\underline{y}_0 \mid \underline{x}_0) \gtrless \ln T$. TYPE B $\lambda_B(\underline{y}_0) + \lambda_A(\underline{x}_0 \mid \underline{y}_0) \gtrless \ln T$	. CENTRAL PROCESSOR USES BOTH OBSERVATION VECTORS $\underline{x}_o$ AND $\underline{y}_o$ DIRECTLY
2. DISTRIBUTED DECISION ALGORITHM A	$\lambda_A(\underline{x}_0) + \lambda_B(\underline{y}_0 \mid \hat{\underline{x}}_1) \gtrless \ln T$ * USED IN PROCESSOR A	. PROCESSOR B ESTIMATES OBSERVATION VECTOR $\underline{x}_o$ BY USING NECESSARY PARAMETERS . THUS PROCESSOR B USES $\underline{y}_o$ AND $\hat{\underline{x}}_{\wedge}$ INSTEAD OF USING $\underline{y}_o$ AND $\underline{x}_o$
3. DISTRIBUTED DECISION ALGORITHM B	$\lambda_B(\underline{y}_0) + \lambda_A(\underline{x}_0 \mid \hat{\underline{y}}_1) \gtrless \ln T$ * USED IN PROCESSOR B	. PROCESSOR A ESTIMATES OBSERVATION VECTOR $\underline{y}_o$ BY USING NECESSARY PARAMETERS . THUS PROCESSOR A USES $\underline{x}_o$ AND $\hat{\underline{y}}_{\wedge}$ INSTEAD OF USING $\underline{x}_o$ AND $\underline{y}_o$

TABLE 2

DIFFERENT VECTORS IN ALGORITHMS

ALGORITHM	USING OBSERVATION VECTORS	
1. CENTRALIZED DECISION ALGORITHM	$\underline{x}_o$ AND $\underline{y}_o$ IN ONE PROCESSOR	
2. DISTRIBUTED DECISION ALGORITHM A	PROCESSOR A	PROCESSOR B
	$\underline{x}_o$	$\underline{y}_o$ and $\hat{\underline{x}}_i$
		$\hat{\underline{x}}_i = \underline{m}_x^{(i)} + \mathbf{B}_{xy}^{(i)} [\mathbf{K}_y^{(i)}]^{-1} [\underline{y} - \underline{m}_y^{(i)}]$
3. DISTRIBUTED DECISION ALGORITHM B	PROCESSOR A	PROCESSOR B
	$\underline{x}_o$ AND $\hat{\underline{y}}_i$	$\underline{y}_o$
	$\hat{\underline{y}}_i = \underline{m}_y^{(i)} + \mathbf{B}_{xy}^{(i)T} [\mathbf{K}_x^{(i)}]^{-1} [\underline{x} - \underline{m}_x^{(i)}]$	

IV. SIMULATION

This chapter contains an evaluation and comparison of distributed decision rules A and B, and the centralized decision rule. The generation of random observation vectors and the calculation of their resulting statistics are discussed in sections A and B. In section C the results of the decision algorithms are compared to the results obtained from classification using a centralized algorithm.

A. RANDOM VECTOR GENERATION

The observation vectors $\underline{x}$ and $\underline{y}$ are generated by using a linear difference equation with white noise excitation. This difference equation can model, for example, the time series of radar cross section values that result when the target is observed by the sensors over a relatively short period of time. If W_1 and W_2 are independent white noise processes this difference equation has the form:

$$\begin{bmatrix} x'(I) \\ y'(I) \end{bmatrix} = A_1 \begin{bmatrix} x'(I-1) \\ y'(I-1) \end{bmatrix} + A_2 \begin{bmatrix} x'(I-2) \\ y'(I-2) \end{bmatrix} + \dots + A_p \begin{bmatrix} x'(I-p) \\ y'(I-p) \end{bmatrix} + \mathbf{K}_w^{\frac{1}{2}} \begin{bmatrix} w_1(I) \\ w_2(I) \end{bmatrix} \tag{4.1}$$

where

$$A_i = \begin{bmatrix} a_x^{(i)} & a_{xy}^{(i)} \\ a_{yx}^{(i)} & a_y^{(i)} \end{bmatrix} \tag{4.2}$$

This generates a pair of time series for x and y that are correlated and have zero mean. The measurements x and y that represent the observations are then defined by:

$$\begin{bmatrix} x(I) \\ y(I) \end{bmatrix} = \begin{bmatrix} x'(I) + m_x \\ y'(I) + m_y \end{bmatrix} \qquad (4.3)$$

where m_x and m_y are the mean values of the observations. The observation vectors $\underline{x}$ and $\underline{y}$ then represent n samples of the time series. In this procedure it is assumed that $[A_i]$ and $[K_w]^{1/2}$ are given in advance and that white noise $W_1(I)$ and $W_2(I)$ have been previously generated and are available in a white noise data file.

The difference equation is implemented by a program with the title "GEN" [Appendix A]. If, for example, the observation vectors $\underline{x}$ and $\underline{y}$ have 32 time points each and a set of 128 independent vectors is needed then the program GEN generates two data sets. Each is an array of size 128 X 32 whose rows represent individual vectors $\underline{x}$ and $\underline{y}$. These data are written to the disk with file names such as "X11", "X12", "Y11", and "Y12" to be used later in the decision test algorithm. In the file name X12 the first number "1" represents test case one, and second number 2 stands for class 2 data.

B. GENERATION OF STATISTICS OF RANDOM VECTORS

After the observation vectors in files X11, Y11, X12, and Y12 are generated, the joint statistics of these vectors are calculated. The statistics are used in the decision algorithms.

Let the dimension of the vectors be N and M and the number of vectors generated be L. Then mean, covariance, and cross covariance parameters are calculated using the following equations:

$$\underline{m}_x = \frac{1}{L} \sum_{k=1}^{L} \underline{x}^{(k)} \tag{4.4}$$

$$\underline{m}_y = \frac{1}{L} \sum_{k=1}^{L} \underline{y}^{(k)} \tag{4.5}$$

$$\mathbf{K}_x = \frac{1}{L} \sum_{k=1}^{L} (\underline{x}^{(k)} - \underline{m}_x) (\underline{x}^{(k)} - \underline{m}_x)^T \tag{4.6}$$

$$\mathbf{K}_y = \frac{1}{L} \sum_{k=1}^{L} (\underline{y}^{(k)} - \underline{m}_y) (\underline{y}^{(k)} - \underline{m}_y)^T \tag{4.7}$$

$$\mathbf{B}_{xy} = \frac{1}{L} \sum_{k=1}^{L} (\underline{x}^{(k)} - \underline{m}_x) (\underline{y}^{(k)} - \underline{m}_y)^T \tag{4.8}$$

Observe that two sets of each of the parameters in Eqs. 4.4 - 4.8 are required: one set for class 1 and one set for

class 2. These calculations are performed by the program "STAT" [Appendix B] and the parameters are written to output files. From the file of vectors X11 the program STAT generates $\underline{m}_x^{(1)}$, and $[K_x^{(1)}]$; from Y11 it produces $\underline{m}_y^{(1)}$ and $[K_y^{(1)}]$; and from both X11 and Y11 it calculates $[B_{xy}^{(1)}]$. These represent the statistical parameters of the class 1 data. The files X12 and Y12 are used in a similar manner to produce $\underline{m}_x^{(2)}$, $[K_x^{(2)}]$, $\underline{m}_y^{(2)}$, $[K_y^{(2)}]$, and $[B_{xy}^{(2)}]$. These represent the statistical parameters of the class 2 data.

C. CLASSIFICATION PROGRAM

When observation vectors and their statistics are available, one can test the distributed classification algorithms and compare their results to the results of the centralized algorithm. A program "DECAL" [Appendix C] was written to implement these decision algorithms. This program has three main parts consisting of distributed decision rule A(denoted simply by "A"), distributed decision rule B(denoted simply by "B"), and the centralized decision rule(denoted simply by "C"). In this program every algorithm computes its own log likelihood ratio statistic to be compared to the threshold value. The statistics corresponding to each pair of observation vectors for each of the decision rules, A, B, and C are written to a disk file and used to compute the correct decision rates.

A Fortran program "ANAL" [Appendix D] generates the varying threshold values that are used with the data generated by DECAL to decide upon the classes of the observed objects. This organization of programs allows us to generate classification results for many threshold values without excessive computation. The threshold values are expressed in terms of the prior probabilities $p_r(\omega_1)$ and $p_r(\omega_2)$ which are chosen so that the condition of "$p_r(\omega_1) + p_r(\omega_2) = 1.0$" is satisfied.

D. CLASSIFICATION EXPERIMENTS

If a correct analysis is performed, one can fit an appropriate time series model to the sensor data to represent the observations made on two distinct types of targets such as those shown in Fig. 4.1.

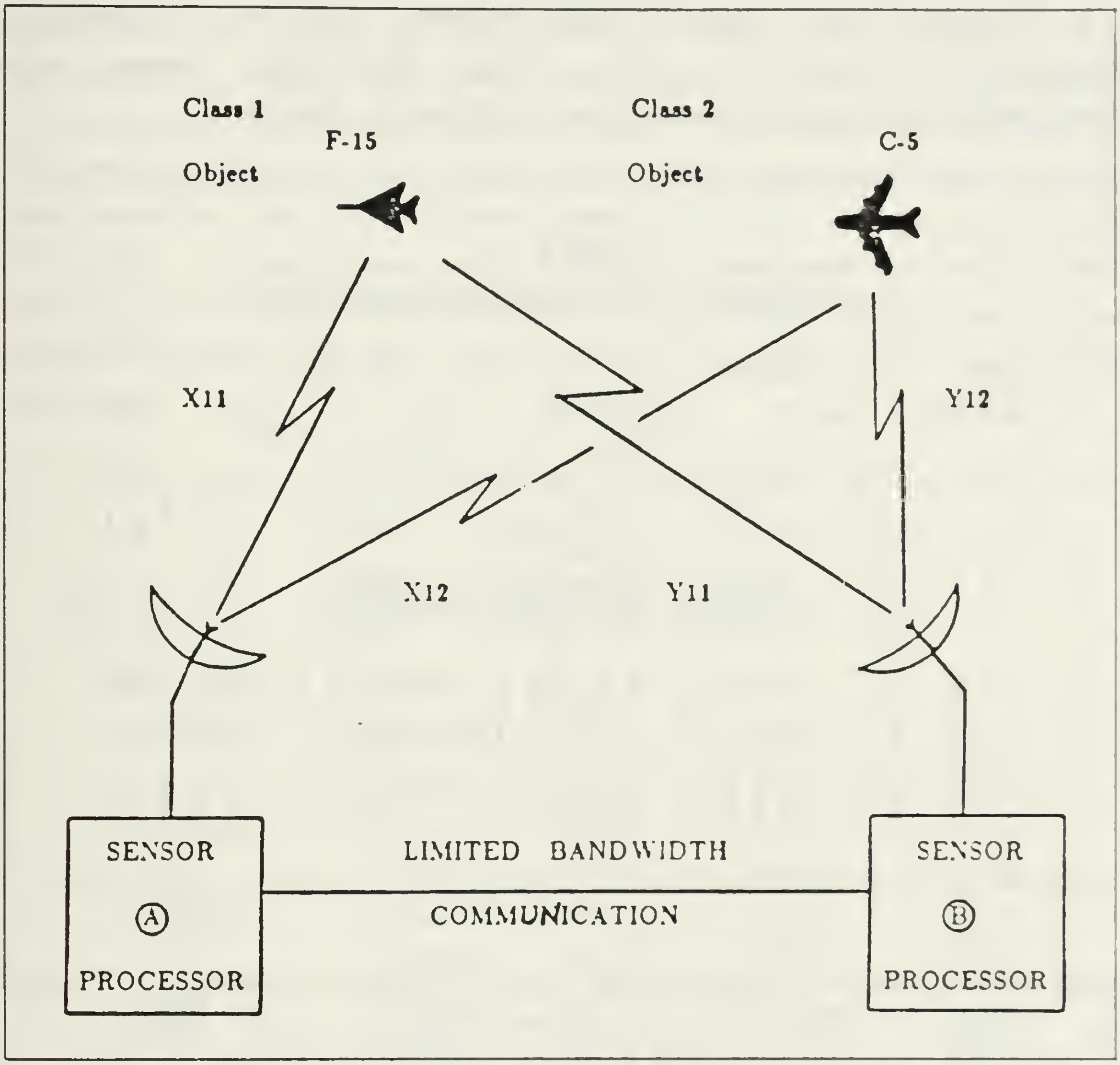

Figure 4.1 Aircraft Type Detection and Observation Vectors

For the analysis here we are more interested in characterizing the distributed decision algorithm

performance for various second moment statistical properties of the observation vectors, such as mean, variance, and correlation. The cases chosen for analysis should not be interpreted to mean that we are attempting to model real target data.

For our experiments, we generated data according to Eqs. 4.1 through 4.3 with the order of the difference equation(p) equal to one. Four different cases were considered; their parameters are given in Table 3.

TABLE 3

PARAMETERS IN DIFFERENCE EQUATIONS

TEST CASE NO	CLASS 1 $[A_1]$	$[K_w]^{1/2}$	M	CLASS 2 $[A_1]$	$[K_w]^{1/2}$	M
1	$\begin{pmatrix} .5 & 0 \\ 0 & .4 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} -.5 & 0 \\ 0 & -.6 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$
2	$\begin{pmatrix} .5 & 0 \\ 0 & .5 \end{pmatrix}$	$\begin{pmatrix} 1 & .2 \\ .2 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} .5 & 0 \\ 0 & .5 \end{pmatrix}$	$\begin{pmatrix} 1 & .2 \\ .2 & 1 \end{pmatrix}$	$\begin{pmatrix} 1.5 \\ 1.5 \end{pmatrix}$
3	$\begin{pmatrix} .5 & .2 \\ .2 & .4 \end{pmatrix}$	$\begin{pmatrix} 1 & .2 \\ .2 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} -.5 & 0 \\ 0 & -.6 \end{pmatrix}$	$\begin{pmatrix} 1 & .2 \\ .2 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$
4	$\begin{pmatrix} .6 & .2 \\ .3 & .5 \end{pmatrix}$	$\begin{pmatrix} 1 & .2 \\ .2 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} .5 & .1 \\ .1 & .4 \end{pmatrix}$	$\begin{pmatrix} 1 & .1 \\ .1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$

Each test case used data from two different classes. In all but case 1 the filter coefficients $[A_1]$ and/or the covariance matrix $[K_w]^{1/2}$ resulted in observation vectors $\underline{x}$ and $\underline{y}$ that are correlated with each other. If the observation vectors $\underline{x}$ and $\underline{y}$ are uncorrelated, the conditional probability density function $p(\underline{y}|\underline{x})$ becomes the same as the unconditional density function $p(\underline{y})$. If this is true for both classes, as in case 1, then the three decision rules A, B, and C will be equivalent.

Several specific cases are represented here. In cases 1 and 3, the class 2 filter has negative A_1 parameters; this makes the time series change very rapidly up and down. Since the data of class 1 does not have this property, we expect that the decision rules can discriminate between the two classes based on the correlation of the time series. In test case 2, class 2 has non-zero mean while class 1 has zero mean. Since the mean values are the only differences, the classification can only be based on these differences in the mean values. In test case 4 the mean values are also non-zero but both the class 1 mean and the class 2 mean are the same. In addition, the filter parameters for each class and the noise covariances are very similar. This makes the classification of the observations a relatively difficult problem.

TABLE 4

CORRECT DECISION RATE(%)
4-DIMENSIONAL 128 VECTORS

TEST CASE	CLASS	A	B	C
CASE #1	CLASS-1	85.9	85.9	85.2
	CLASS-2	84.4	85.2	82.8
CASE #2	CLASS-1	93.0	93.8	92.2
	CLASS-2	85.2	85.9	89.1
CASE #3	CLASS-1	81.3	83.6	85.2
	CLASS-2	85.9	86.7	85.9
CASE #4	CLASS-1	85.9	87.5	57.0
	CLASS-2	19.5	17.2	60.2

The results of classification for these test cases is shown in Tables 4 and 5. The results are based on a threshold corresponding to equal prior probabilities. The

first test set was 4-dimensional (i.e. $\underline{x}$ and $\underline{y}$ each consisted of four time samples) and consisted of 128 pairs of observation vectors $\underline{x}$ and $\underline{y}$. These results are given in Table 4. Most of the results show probabilities of correct classification in the range of about 85 to 90 percent. For test case 4 the probability of correct classification achieved by decision rules A and B is quite high for class 1 but very low for class 2. However, if the classifier threshold is adjusted by choosing different prior probabilities, the results are similar (but slightly worse) than the results for the centralized rule C. (The reader may refer to Appendix E.)

TABLE 5

CORRECT DECISION RATE(%)
32-DIMENSIONAL 128 VECTORS

TEST CASE	CLASS	A	B	C
CASE #1	CLASS-1	100.	100.	100.
	CLASS-2	100.	100.	100.
CASE #2	CLASS-1	100.	100.	100.
	CLASS-2	100.	100.	100.
CASE #3	CLASS-1	100.	100.	100.
	CLASS-2	99.2	99.2	93.8
CASE #4	CLASS-1	100.	100.	88.3
	CLASS-2	13.3	6.3	91.4

The second test set was 32-dimensional and again consisted of 128 observation vectors $\underline{x}$ and $\underline{y}$. The results are given in Table 5. Note that in cases 1, 2, and 3 all vectors were classified correctly. That shows that the classes are easily separated by any of the decision rules if 32 time samples are used.

In test case 4, the degraded performance is explained by the parameters in Table 3. Here both classes have similar correlation parameters, and both mean values are identical. This case was designed to be the most difficult.

By varying the prior probabilities one can change the threshold in the decision algorithms and therefore trade off the probability of correct classification of one class for incorrect classification of the other class. A graph of these probabilities is known as an "operating characteristic" for the decision rule. The results in Tables 4 and 5 represent a single point on each of the operating characteristics. Operating characteristics for cases 1,2,3, and 4 of Table 4 and case 4 of Table 5 are given in Figs. 4.2 through 4.5. The three different types of lines in the graph represent the results of the three different algorithms. These results are also given as tables in the Appendices. The correct decision rate is shown in the output data "GRAPH4" [Appendix E] for the 4-dimensional cases and "GRAPH32" [Appendix F] for the 32-dimensional cases.

It is interesting to note that in most cases the performance of the distributed decision rules compared favorably to that of the centralized decision rule. It is also interesing to note that the performance of decision rules A and B was always close together although the data in the test cases exhibited no symmetry in their defining parameters.

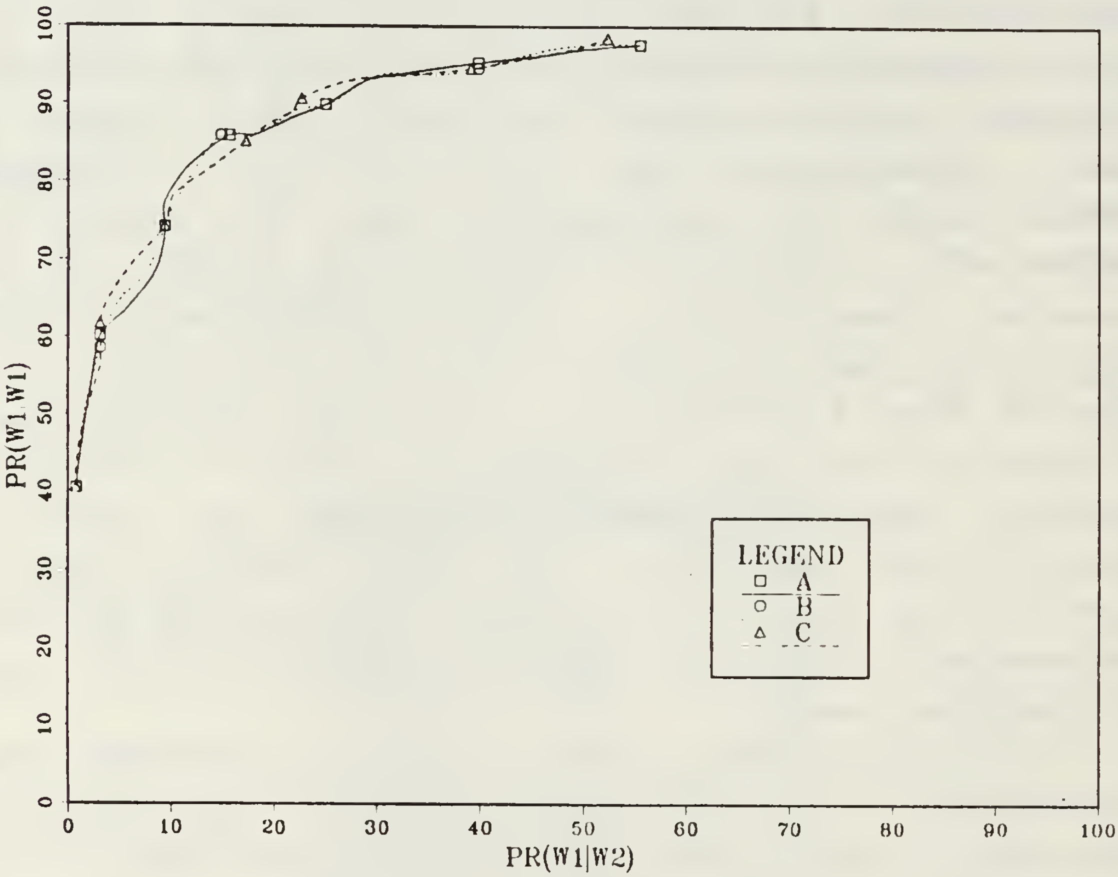

Figure 4.2 Operating Characteristics Graph of Test Case 1(4-Dimensional Vectors)

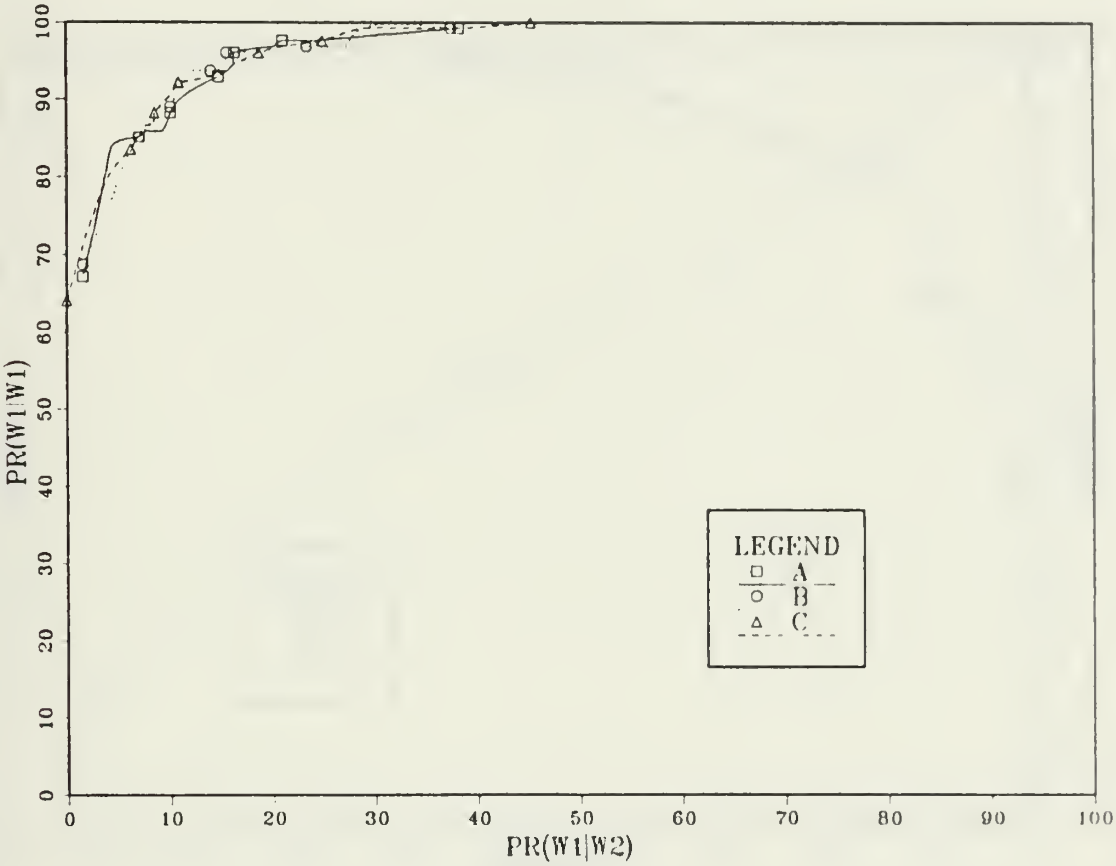

Figure 4.3 Operating Characteristics Graph of Test Case 2(4-Dimensional Vectors)

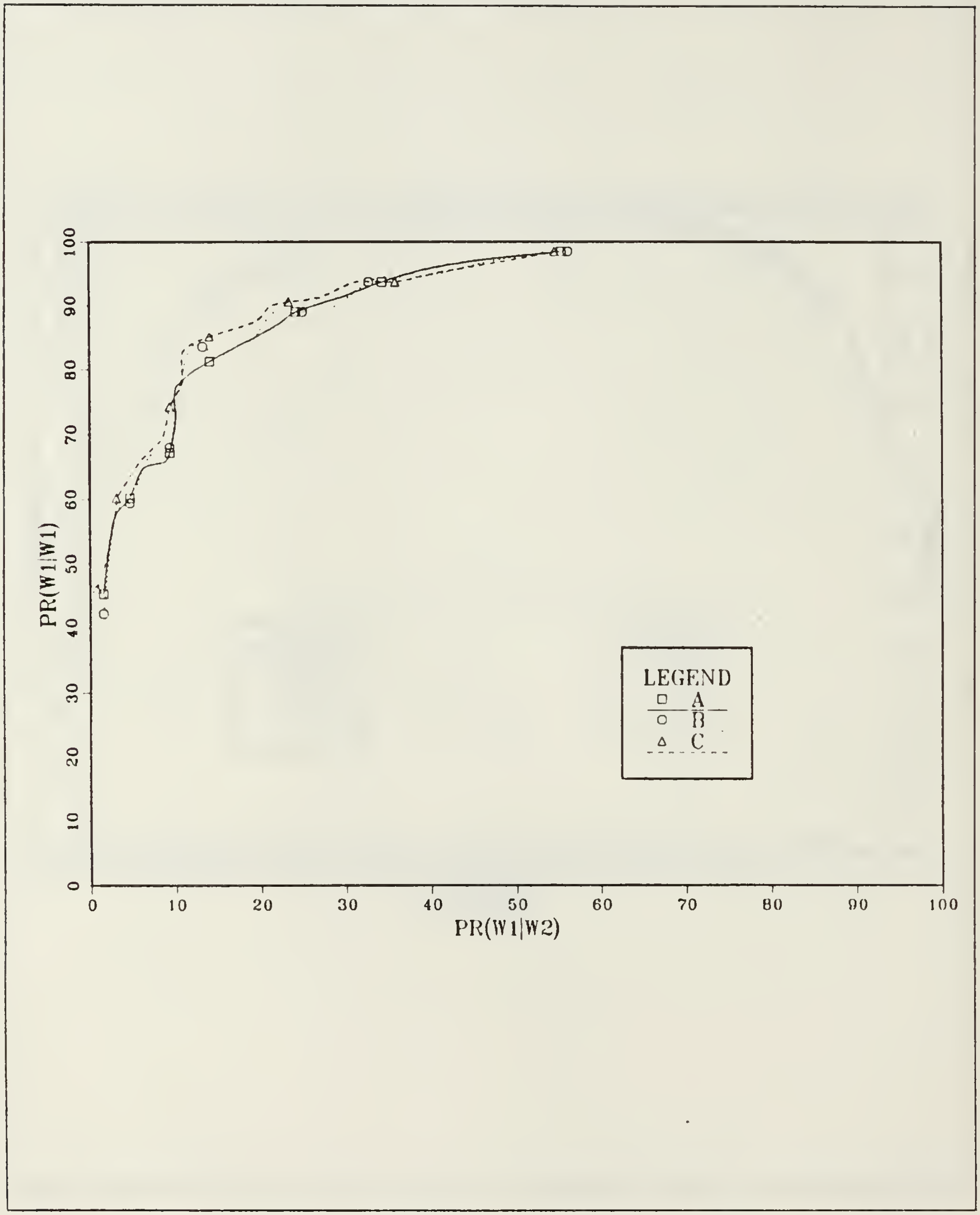

Figure 4.4 Operating Characteristics Graph
of Test Case 3(4-Dimensional Vectors)

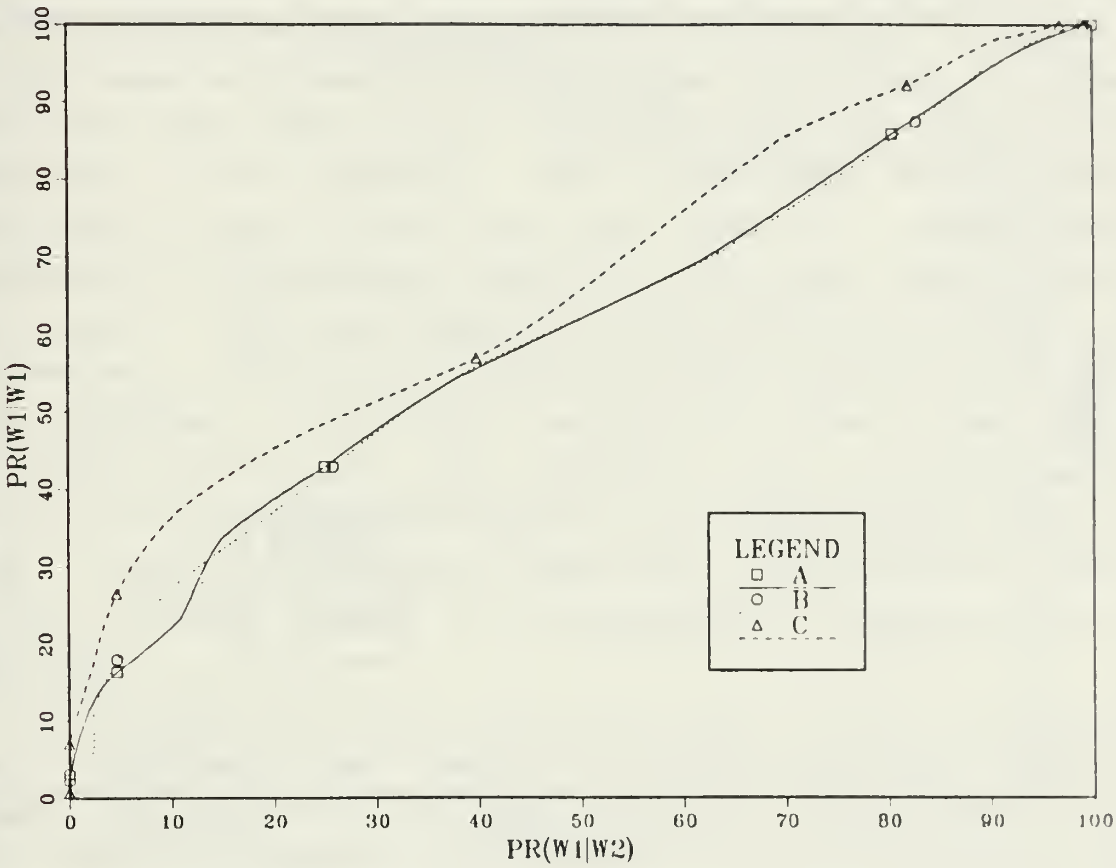

Figure 4.5 Operating Characteristics Graph of Test Case 4(4-Dimensional Vectors)

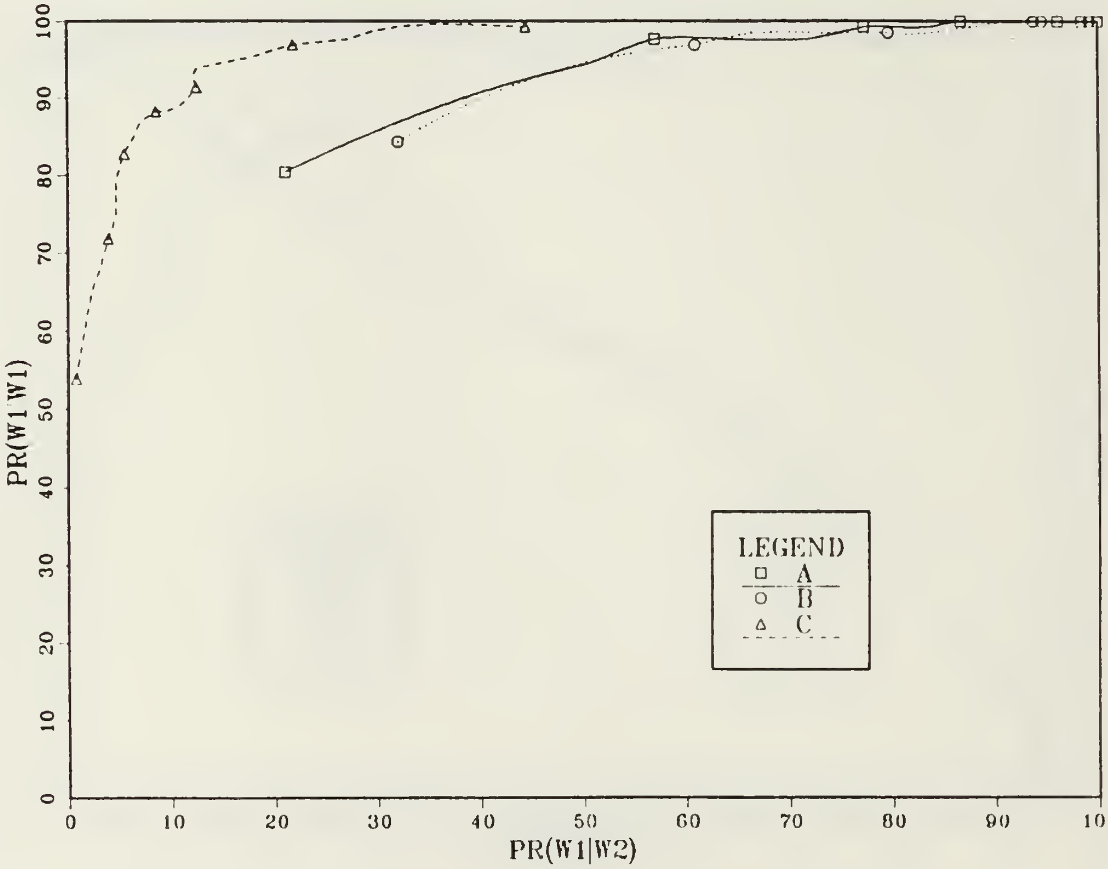

Figure 4.6 Operating Characteristics Graph of Test Case 4(32-Dimensional vectors)

V. CONCLUSIONS

The specific goals were all met in this thesis. The distributed decision rules were introduced and compared to the centralized decision rule. Since only one observation vector (either x or y) is available in each processor, the results of the distributed decision rule can not in general be the same as those of a centralized decision rule. The decision algorithms were explained mathematically and compared to one another. The difference between the algorithms arises from the fact that one sensor must estimate the observation vector of the other sensor using the locally measured observation vector and all available parameters. Simulation experiments for a number of cases with different statistical properties showed that when multiple observations are involved, the two distributed decision rules compare favorably to the centralized decision rule. Even when the vectors have high dimensionality, only a fixed limited amount of interprocessor communication is required.

In the two distributed decision rules, if each processor has a different class decision for the commonly observed object, an ambiguous situation results. In this case, one can either disregard that decision or use the following method. By comparing each log likelihood ratio statistic to the threshold value, one can select the decision which is further from the threshold value. This procedure is intuitively reasonable because decisions made when the statistic is close to the threshold value(observations in the region near the decision boundary) are more likely to be incorrect.

Further research may center on analytical characterization of these distributed decision rules and further analysis of the situation where the two rules A and B do not agree.

APPENDIX A

GEN FORTRAN

```
c     This program generates two sets of random observation
c     vectors i.e. X11 and Y11.

      REAL*8 A(9,2,2),MX,MY,XP(32),YP(32),X(32),
     +       Y(32),KW(2,2),W1(32),W2(32)

      INTEGER H,I,J,K,L,M,N,P

      N=32
      M=32
      P=1

      READ(2,*) MX,MY
      READ(2,*) (((A(I,J,K),K=1,2),J=1,2),I=1,P)
      READ(2,*) ((KW(I,J),J=1,2),I=1,2)

10    READ(3,*,END=50) (W1(I),I=1,N)
      READ(3,*) (W2(I),I=1,N)

      XP(1)=KW(1,1)*W1(1)+KW(1,2)*W2(1)
      YP(1)=KW(2,1)*W1(1)+KW(2,2)*W2(1)

      DO 30 I=2,N
            XP(I)=0.
            YP(I)=0.
            K=I
            IF (I.GT.P) K=P
            L=I-1
         DO 20 J=1,K
            XP(I)=XP(I)+A(J,1,1)*XP(L)+A(J,1,2)*YP(L)

            YP(I)=YP(I)+A(J,2,1)*XP(L)+A(J,2,2)*YP(L)
            L = L-1
20    CONTINUE
         XP(I)= XP(I) + KW(1,1)*W1(I) + KW(1,2)*W2(I)
         YP(I)= YP(I) + KW(2,1)*W1(I) + KW(2,2)*W2(I)
30    CONTINUE


      DO 40 I=1,N
         X(I)=XP(I)+MX
         Y(I)=YP(I)+MY
40    CONTINUE
      WRITE (7,41) (X(I),I=1,N)
41    FORMAT(1X,4(2X,E15.8))
      WRITE (8,42) (Y(I),I=1,M)
42    FORMAT(1X,4(2X,E15.8))

      GO TO 10
50    STOP
      END
```

APPENDIX B

STAT FORTRAN

```
C     This program computes all the necessary
c     parameters of the given sets of vectors
c     i.e. X11 and Y11, or X12 and Y12.
c     Matrix manipulation subroutines are
c     from the IMSL library [Ref. 5].

      REAL*8 MX(32),MY(32),XP(32),YP(32),
     +       X(32),Y(32),KW(32,32),
     +       KX(32,32),KY(32,32),BXY(32,32),
     +       XD(32,128),YD(32,128),
     +       SKX(32,32),SKY(32,32),SBXY(32,32)
c
      INTEGER I,J,K,L,M,N,IER
c
      L=128
      M=32
      N=32
c
         READ(2,*,END=05) ((XD(I,J),J=1,N),I=1,L)
         READ(3,*) ((YD(I,J),J=1,M),I=1,L)
c       WRITE(8,*) ((XD(I,J),J=1,N),I=1,L)
c       WRITE(9,*) ((YD(I,J),J=1,M),I=1,L)
c
05    MX(I)=0.
      MY(I)=0.
c
      DO 20 I=1,N
         DO 10 J=1,L
            MX(I)=MX(I)+XD(J,I)
            MY(I)=MY(I)+YD(J,I)
10    CONTINUE
            MX(I)=1./L*MX(I)
            MY(I)=1./L*MY(I)
20    CONTINUE
c
      DO 23 I=1,N
         DO 23 J=1,N
            SKX(I,J)=0.
            SKY(I,J)=0.
            SBXY(I,J)=0.
23    CONTINUE
c
25       READ(4,*,END=35) (X(I),I=1,N)
         READ(5,*) (Y(I),I=1,M)
c       WRITE(6,*) (X(I),I=1,N)
c       WRITE(7,*) (Y(I),I=1,M)
c
         DO 27 I=1,N
            X(I)=X(I)-MX(I)
            Y(I)=Y(I)-MX(I)
27    CONTINUE
c
      CALL VMULFP(X,X,N,1,N,N,N,KX,N,IER)
      CALL VMULFP(Y,Y,M,1,M,M,M,KY,M,IER)
      CALL VMULFP(X,Y,N,1,M,N,M,BXY,N,IER)
```

```
c
      DO 30 I=1,N
         DO 30 J=1,N
            SKX(I,J)=SKX(I,J)+KX(I,J)
            SKY(I,J)=SKY(I,J)+KY(I,J)
            SBXY(I,J)=SBXY(I,J)+BXY(I,J)
30    CONTINUE
c
      GO TO 25
c
35    DO 40 I=1,N
         DO 40 J=1,N
            KX(I,J)=1./L*SKX(I,J)
            KY(I,J)=1./L*SKY(I,J)
            BXY(I,J)=1./L*SBXY(I,J)
40    CONTINUE
c
      WRITE(7,41) (MX(I),I=1,N)
41    FORMAT (1X,4(2X,E15.8))
      WRITE(7,42) (MY(I),I=1,M)
42    FORMAT (1X,4(2X,E15.8))
      WRITE(7,43)((KX(I,J),J=1,N),I=1,N)
43    FORMAT (1X,4(2X,E15.8))
      WRITE(7,44)((KY(I,J),J=1,M),I=1,M)
44    FORMAT (1X,4(2X,E15.8))
      WRITE(7,45)((BXY(I,J),J=1,M),I=1,N)
45    FORMAT (1X,4(2X,E15.8))
c
      STOP
      END
```

APPENDIX C

DECAL FORTRAN

```
c    This program computes the final scalar values
c    of three different algorithms which will be
c    compared with the threshold value.
c    Matrix manipulation subroutines are from
c    the IMSL library [Ref. 5].

c           ****************************************
c************   DECLARATIONS FOR DIST. RULE A   *********
c           ****************************************
c
      REAL*8  RX,RPY,SUM1,SUM2,SUM3,SUM4,SUM5,
     +        PRW1,PRW2,T,VAL,
     +        DKX1,DKX2,DKY1,DKY2,DKYX1,DKYX2,
     +        MIM1,MIM2,C1,C2,
c
c
     +        X(32),MX1(32),MX2(32),
     +        Y(32),MY1(32),MY2(32),
     +        MK1(32),MK2(32),
     +        MB1(32),MB2(32),
     +        B1MY(32),B2MY(32),
c
c
     +        IMB1(32),IMB2(32),BIM1(32),BIM2(32),
     +        MBI1(32),MBI2(32),MIB1(32),MIB2(32)
c
c
      REAL*8  A1(32,32),B1(32),
     +        A2(32,32),B2(32),
c
c
     +        WKAREA(1160),
     +        KX1(32,32),IKX1(32,32),KX1D(32,32),
     +        KX2(32,32),IKX2(32,32),KX2D(32,32),
     +        KY1(32,32),IKY1(32,32),KY1D(32,32),
     +        KY2(32,32),IKY2(32,32),KY2D(32,32),
     +        BXY1(32,32),
     +        BXY2(32,32),
     +        KYX1(32,32),IKYX1(32,32),KYX1D(32,32),
     +        KYX2(32,32),IKYX2(32,32),KYX2D(32,32)
c
c
      REAL*8  BB1X(32,32),BB2X(32,32),
     +        BX1Y(32,32),BX2Y(32,32),
     +        BY1(32,32),BY2(32,32),
c
c
     +        BKB1(32,32),BKB2(32,32),
     +        IBY1(32,32),IBY2(32,32),
     +        BYI1(32,32),BYI2(32,32),
     +        BIB1(32,32),BIB2(32,32)

c
c
      INTEGER  I,J,L,M,N,
```

```
+            IA,IDGT,IER,CLASS,
+            NCL1,NCL2,NCLA1,NCLA2,NCLAS1,NCLAS2
c        ***************************
c******* DECLARATION FOR DIST. B *********
c        ***************************
      REAL*8 BKX1(32,32),BKX2(32,32),
+            BX1(32,32),BX2(32,32),
+            KXY1(32,32),KXY2(32,32),
+            KXY1D(32,32),KXY2D(32,32),
+            IKXY1(32,32),IKXY2(32,32),
+            IBX1(32,32),IBX2(32,32),
+            BXI1(32,32),BXI2(32,32),
+            BIX1(32,32),BIX2(32,32),
+            A3(32,32),A4(32,32),
c
+            MIX1(32),MIX2(32),          B3(32),B4(32),
+            BIP1(32),BIP2(32),          MBX1(32),MBX2(32),
+            B1MX(32),B2MX(32),          MB1X(32),MB2X(32),
+            MKY1(32),MKY2(32),          IXB1(32),IXB2(32)
c
      REAL*8 DKXY1,DKXY2,MXM1,MXM2,      C3,C4,
+            SUM11,SUM12,SUM13,SUM14,SUM15,
+            RY,RPX,VA
c
      INTEGER CLA
c
c              ********************************
c************** DECLARATION FOR CENTRALIZED. *********
c              ********************************
c
c
      REAL*8 A5(32,32),B5(32),
+            XMX1(32),XMX2(32),       BYT1(32),BYT2(32),
+            MBT1(32),MBT2(32),       KBT1(32),KBT2(32),
+            MBK1(32),MBK2(32),
c
+            MT1,MT2,RBY,C5,SUM24,SUM25,V
c
      INTEGER CL,COUNT
c
c  INITIALIZATION!!!!
c
      NCL1=0
      NCL2=0
      NCLA1=0
      NCLA2=0
      NCLAS1=0
      NCLAS2=0
c
      L=0
      M=32
      N=32
      IDGT=4
c
c     PRW2=.5
c     PRW1=.5
c     T=DLOG(PRW1/PRW2)
c       WRITE(7,*)'T=',T
c
cINPUT PARAMETERS!!!!!
c
      READ(2,*)(MX1(I),I=1,N)
c       WRITE(7,*)(MX1(I),I=1,N)
      READ(2,*)(MY1(I),I=1,M)
c       WRITE(7,*)(MY1(I),I=1,N)
      READ(2,*)((KX1(I,J),J=1,N),I=1,N)
c       WRITE(7,*)((KX1(I,J),J=1,N),I=1,N)
      READ(2,*)((KY1(I,J),J=1,M),I=1,M)
c       WRITE(7,*)((KY1(I,J),J=1,M),I=1,M)
      READ(2,*)((BXY1(I,J),J=1,M),I=1,N)
```

```
c       WRITE(7,*)((BXY1(I,J),J=1,M),I=1,N)
c
c
      READ(3,*)(MX2(I),I=1,N)
c       WRITE(7,*)(MX2(I),I=1,N)
      READ(3,*)(MY2(I),I=1,M)
c       WRITE(7,*)(MY2(I),I=1,M)
      READ(3,*)((KX2(I,J),J=1,N),I=1,N)
c       WRITE(7,*)((KX2(I,J),J=1,N),I=1,N)
      READ(3,*)((KY2(I,J),J=1,M),I=1,M)
c       WRITE(7,*)((KY2(I,J),J=1,M),I=1,M)
      READ(3,*)((BXY2(I,J),J=1,M),I=1,N)
c       WRITE(7,*)((BXY2(I,J),J=1,M),I=1,N)
c
c
c              ************************
c************  DISTRIBUTED RULE A  *************
c              ************************
c
c
c
c
      DO 01 I=1,N
         DO 01 J=1,N
            KX1D(I,J)=KX1(I,J)
            KX2D(I,J)=KX2(I,J)
01    CONTINUE
c       WRITE(7,*)((KX1D(I,J),J=1,N),I=1,N)
c       WRITE(7,*)((KX2D(I,J),J=1,N),I=1,N)
c
c
      DO 02 I=1,M
         DO 02 J=1,M
            KY1D(I,J)=KY1(I,J)
            KY2D(I,J)=KY2(I,J)
02    CONTINUE
c
cSUBROUTINES!!!!!!
c
      CALL LINV2F (KX1,N,N,IKX1,IDGT,WKAREA,IER)
c        WRITE(7,*)((IKX1(I,J),J=1,N),I=1,N)
c        WRITE(7,*)((KX1(I,J),J=1,N),I=1,N)
      CALL LINV2F (KX2,N,N,IKX2,IDGT,WKAREA,IER)
c        WRITE(7,*)((IKX2(I,J),J=1,N),I=1,N)
c
      CALL LINV2F (KY1,M,M,IKY1,IDGT,WKAREA,IER)
c        WRITE(7,*)((IKY1(I,J),J=1,M),I=1,M)
      CALL LINV2F (KY2,M,M,IKY2,IDGT,WKAREA,IER)
c        WRITE(7,*)((IKY2(I,J),J=1,M),I=1,M)
c
      CALL DTERM (N,KX1D,DKX1,N)
c        WRITE(7,*)'DKX1=',DKX1
      CALL DTERM (N,KX2D,DKX2,N)
c        WRITE(7,*)'DKX2=',DKX2
c
      CALL DTERM (M,KY1D,DKY1,M)
c        WRITE(7,*)'DKY1=',DKY1
      CALL DTERM (M,KY2D,DKY2,M)
c        WRITE(7,*)'DKY2=',DKY2
c
      CALL VMULFM (BXY1,IKX1,N,M,N,N,N,BB1X,M,IER)
      CALL VMULFF (BXY1,IKY1,N,M,M,N,M,BX1Y,N,IER)
c
c        WRITE(7,*)((IKX1(I,J),J=1,N),I=1,N)
c        WRITE(7,*)((IKY1(I,J),J=1,M),I=1,M)
c       WRITE(7,*)((BXY1(I,J),J=1,M),I=1,N)
c        WRITE(7,*)((BB1X(I,J),J=1,N),I=1,M)
c        WRITE(7,*)((BX1Y(I,J),J=1,M),I=1,N)
c
c
      CALL VMULFM (BXY2,IKX2,N,M,N,N,N,BB2X,M,IER)
```

```
      CALL VMULFF (BXY2,IKY2,N,M,M,N,M,BX2Y,N,IER)
c
c        WRITE(7,*)((IKX2(I,J),J=1,N),I=1,N)
c        WRITE(7,*)((IKY2(I,J),J=1,M),I=1,M)
c        WRITE(7,*)((BXY2(I,J),J=1,M),I=1,N)
c        WRITE(7,*)((BB2X(I,J),J=1,N),I=1,M)
c        WRITE(7,*)((BX2Y(I,J),J=1,M),I=1,N)
c
      CALL VMULFF (BB1X,BXY1,M,N,M,M,N,BKB1,M,IER)
      CALL VMULFF (BB2X,BXY2,M,N,M,M,N,BKB2,M,IER)
c
c        WRITE(7,*)((BKB1(I,J),J=1,M),I=1,M)
c        WRITE(7,*)((BKB2(I,J),J=1,M),I=1,M)
c
      CALL VMULFF (BB1X,BX1Y,M,N,M,M,N,BY1,M,IER)
      CALL VMULFF (BB2X,BX2Y,M,N,M,M,N,BY2,M,IER)
c
c        WRITE(7,*)((BY1(I,J),J=1,M),I=1,M)
c        WRITE(7,*)((BY2(I,J),J=1,M),I=1,M)
c
      CALL VMULFF (BY1,MY1,M,M,1,M,M,B1MY,M,IER)
      CALL VMULFF (BY2,MY2,M,M,1,M,M,B2MY,M,IER)
c
c        WRITE(7,*)(B1MY(I),I=1,M)
c        WRITE(7,*)(B2MY(I),I=1,M)
c
      DO 10 I=1,M
         MB1(I)=MY1(I)-B1MY(I)
         MB2(I)=MY2(I)-B2MY(I)
10    CONTINUE
c
c        WRITE(7,*)(MB1(I),I=1,M)
c        WRITE(7,*)(MB2(I),I=1,M)
c
      CALL VMULFF (MX1,IKX1,1,N,N,1,N,MK1,1,IER)
      CALL VMULFF (MX2,IKX2,1,N,N,1,N,MK2,1,IER)
c
c        WRITE(7,*)(MK1(I),I=1,M)
c        WRITE(7,*)(MK2(I),I=1,M)
c
      DO 20 I=1,M
         DO 20 J=1,M
            KYX1(I,J)=KY1(I,J)-BKB1(I,J)
            KYX2(I,J)=KY2(I,J)-BKB2(I,J)
c
            KYX1D(I,J)=KYX1(I,J)
            KYX2D(I,J)=KYX2(I,J)
20    CONTINUE
c
c        WRITE(7,*)((KYX1(I,J),J=1,M),I=1,M)
c        WRITE(7,*)((KYX2(I,J),J=1,M),I=1,M)
c
      CALL LINV2F (KYX1,M,M,IKYX1,IDGT,WKAREA,IER)
      CALL LINV2F (KYX2,M,M,IKYX2,IDGT,WKAREA,IER)
c
c        WRITE(7,*)((IKYX1(I,J),J=1,M),I=1,M)
c        WRITE(7,*)((IKYX2(I,J),J=1,M),I=1,M)
c
      CALL DTERM (M,KYX1D,DKYX1,M)
      CALL DTERM (M,KYX2D,DKYX2,M)
c
c        WRITE(7,*)'DKYX1=',DKYX1
c        WRITE(7,*)'DKYX2=',DKYX2
c
      CALL VMULFF (IKYX1,BY1,M,M,M,M,M,IBY1,M,IER)
      CALL VMULFF (IKYX2,BY2,M,M,M,M,M,IBY2,M,IER)
c
c        WRITE(7,*)((IBY1(I,J),J=1,M),I=1,M)
c        WRITE(7,*)((IBY2(I,J),J=1,M),I=1,M)
c
```

```
      CALL VMULFF (IKYX1,MB1,M,M,1,M,M,IMB1,M,IER)
      CALL VMULFF (IKYX2,MB2,M,M,1,M,M,IMB2,M,IER)
c
c        WRITE(7,*)(IMB1(I),I=1,M)
c        WRITE(7,*)(IMB2(I),I=1,M)
c
      CALL VMULFM (BY1,IKYX1,M,M,M,M,M,BYI1,M,IER)
      CALL VMULFM (BY2,IKYX2,M,M,M,M,M,BYI2,M,IER)
c
c        WRITE(7,*)((BYI1(I,J),J=1,M),I=1,M)
c        WRITE(7,*)((BYI2(I,J),J=1,M),I=1,M)
c
      CALL VMULFF (BYI1,BY1,M,M,M,M,M,BIB1,M,IER)
      CALL VMULFF (BYI2,BY2,M,M,M,M,M,BIB2,M,IER)
c
c        WRITE(7,*)((BIB1(I,J),J=1,M),I=1,M)
c        WRITE(7,*)((BIB2(I,J),J=1,M),I=1,M)
c
      CALL VMULFF (BYI1,MB1,M,M,1,M,M,BIM1,M,IER)
      CALL VMULFF (BYI2,MB2,M,M,1,M,M,BIM2,M,IER)
c
c        WRITE(7,*)(BIM1(I),I=1,M)
c        WRITE(7,*)(BIM2(I),I=1,M)
c
      CALL VMULFM (MB1,IKYX1,M,1,M,M,M,MBI1,1,IER)
      CALL VMULFM (MB2,IKYX2,M,1,M,M,M,MBI2,1,IER)
c
c        WRITE(7,*)(MBI1(I),I=1,M)
c        WRITE(7,*)(MBI2(I),I=1,M)
c
      CALL VMULFF (MBI1,BY1,1,M,M,1,M,MIB1,1,IER)
      CALL VMULFF (MBI2,BY2,1,M,M,1,M,MIB2,1,IER)
c
c        WRITE(7,*)(MIB1(I),I=1,M)
c        WRITE(7,*)(MIB2(I),I=1,M)
c
      CALL VMULFF (MBI1,MB1,1,M,1,1,M,MIM1,1,IER)
      CALL VMULFF (MBI2,MB2,1,M,1,1,M,MIM2,1,IER)
c
c        WRITE(7,*)'MIM1=',MIM1
c        WRITE(7,*)'MIM2=',MIM2
c
      DO 30 I=1,N
           B1(I)=2.*(MK1(I)-MK2(I))
         DO 30 J=1,N
           A1(I,J)=IKX2(I,J)-IKX1(I,J)
           SUM3=SUM3+MX2(I)*IKX2(I,J)*MX2(J)
     +           -MX1(I)*IKX1(I,J)*MX1(J)
30    CONTINUE
c
c
c        WRITE(7,*)((A1(I,J),J=1,N),I=1,N)
c        WRITE(7,*)(B1(I),I=1,N)
c
c
           C1=SUM3+DLOG(DKX2/DKX1)
c        WRITE(7,*)'C1=',C1
c
      DO 40 I=1,M
           B2(I)=BIM2(I)+MIB2(I)-IMB2(I)-MBI2(I)
     +           -(BIM1(I)+MIB1(I)-IMB1(I)-MBI1(I))
         DO 40 J=1,M
           A2(I,J)=IKYX2(I,J)-IBY2(I,J)
     +           -BYI2(I,J)+BIB2(I,J)
     +           -(IKYX1(I,J)-IBY1(I,J)
     +           -BYI1(I,J)+BIB1(I,J))
40    CONTINUE
           C2=MIM2-MIM1+DLOG(DKYX2/DKYX1)
c
c        WRITE(7,*)((A2(I,J),J=1,N),I=1,N)
```

```
c          WRITE(7,*)(B2(I),I=1,N)
c          WRITE(7,*) 'C2=',C2
c
45      READ(4,*,END=299)(X(I),I=1,N)
        READ(5,*)(Y(I),I=1,M)
c
        L=L+1
c          WRITE(7,*) (X(I),I=1,N)
c          WRITE(7,*) (Y(I),I=1,M)
c
        SUM1=0.
        SUM2=0.
        SUM3=0.
        SUM4=0.
        SUM5=0.
        SUM11=0.
        SUM12=0.
        SUM13=0.
        SUM14=0.
        SUM15=0.
        SUM24=0.
        SUM25=0.
c
        DO 50 I=1,N
              SUM2=SUM2+B1(I)*X(I)
           DO 50 J=1,N
              SUM1=SUM1+X(I)*A1(I,J)*X(J)
50      CONTINUE
        RX=0.5*(SUM1+SUM2+C1)
c
c
        DO 60 I=1,M
              SUM5=SUM5+B2(I)*Y(I)
           DO 60 J=1,M
              SUM4=SUM4+Y(I)*A2(I,J)*Y(J)
60      CONTINUE
        RPY=0.5*(SUM4+SUM5+C2)
c
c
        VAL=RX+RPY
c
c
        IF(VAL.GT.T) THEN
             CLASS=1
             NCLAS1=NCLAS1+1
        ELSE
             CLASS=2
             NCLAS2=NCLAS2+1
        END IF
c
c
c
c                     ************************
c***************  DISTRIBUTED RULE B *******************
c                     ************************
cSUBROUTINES!!!!!!
c
c
c
100     CALL VMULFP (BX1Y,BXY1,N,M,N,N,N,BKX1,N,IER)
        CALL VMULFP (BX2Y,BXY2,N,M,N,N,N,BKX2,N,IER)
c
c
        CALL VMULFF (BX1Y,BB1X,N,M,N,N,M,BX1,N,IER)
        CALL VMULFF (BX2Y,BB2X,N,M,N,N,M,BX2,N,IER)
c
c
        CALL VMULFF (BX1,MX1,N,N,1,N,N,B1MX,N,IER)
        CALL VMULFF (BX2,MX2,N,N,1,N,N,B2MX,N,IER)
c
```

```
C
      DO 110 I=1,N
         MB1X(I)=MX1(I)-B1MX(I)
         MB2X(I)=MX2(I)-B2MX(I)
110   CONTINUE
C
C
      CALL VMULFF (MY1,IKY1,1,M,M,1,M,MKY1,1,IER)
      CALL VMULFF (MY2,IKY2,1,M,M,1,M,MKY2,1,IER)
C
C
      DO 120 I=1,N
         DO 120 J=1,N
            KXY1(I,J)=KX1(I,J)-BKX1(I,J)
            KXY2(I,J)=KX2(I,J)-BKX2(I,J)
C
            KXY1D(I,J)=KXY1(I,J)
            KXY2D(I,J)=KXY2(I,J)
120   CONTINUE
C
C
      CALL LINV2F (KXY1,N,N,IKXY1,IDGT,WKAREA,IER)
      CALL LINV2F (KXY2,N,N,IKXY2,IDGT,WKAREA,IER)
C
C
      CALL DTERM (N,KXY1D,DKXY1,N)
      CALL DTERM (N,KXY2D,DKXY2,N)
C
C
      CALL VMULFF (IKXY1,BX1,N,N,N,N,N,IBX1,N,IER)
      CALL VMULFF (IKXY2,BX2,N,N,N,N,N,IBX2,N,IER)
C
C
      CALL VMULFF (IKXY1,MB1X,N,N,1,N,N,IXB1,N,IER)
      CALL VMULFF (IKXY2,MB2X,N,N,1,N,N,IXB2,N,IER)
C
C
      CALL VMULFM (BX1,IKXY1,N,N,N,N,N,BXI1,N,IER)
      CALL VMULFM (BX2,IKXY2,N,N,N,N,N,BXI2,N,IER)
C
C
      CALL VMULFF (BXI1,BX1,N,N,N,N,N,BIX1,N,IER)
      CALL VMULFF (BXI2,BX2,N,N,N,N,N,BIX2,N,IER)
C
C
      CALL VMULFF (BXI1,MB1X,N,N,1,N,N,BIP1,N,IER)
      CALL VMULFF (BXI2,MB2X,N,N,1,N,N,BIP2,N,IER)
C
C
      CALL VMULFM (MB1X,IKXY1,N,1,N,N,N,MBX1,1,IER)
      CALL VMULFM (MB2X,IKXY2,N,1,N,N,N,MBX2,1,IER)
C
C
      CALL VMULFF (MBX1,BX1,1,N,N,1,N,MIX1,1,IER)
      CALL VMULFF (MBX2,BX2,1,N,N,1,N,MIX2,1,IER)
C
C
      CALL VMULFF (MBX1,MB1X,1,N,1,1,N,MXM1,1,IER)
      CALL VMULFF (MBX2,MB2X,1,N,1,1,N,MXM2,1,IER)
C
C
      DO 130 I=1,M
            B4(I)=2.*(MKY1(I)-MKY2(I))
         DO 130 J=1,M
            A4(I,J)=IKY2(I,J)-IKY1(I,J)
            SUM13=SUM13+MY2(I)*IKY2(I,J)*MY2(J)
     +               -MY1(I)*IKY1(I,J)*MY1(J)
130   CONTINUE
            C4=SUM13+DLOG(DKY2/DKY1)
```

```
      DO 140 I=1,N
            B3(I)=BIP2(I)+MIX2(I)-IXB2(I)-MBX2(I)
     +            -(BIP1(I)+MIX1(I)-IXB1(I)-MBX1(I))
         DO 140 J=1,N
            A3(I,J)=IKXY2(I,J)-IBX2(I,J)
     +             -BXI2(I,J)+BIX2(I,J)
     +             -(IKXY1(I,J)-IBX1(I,J)
     +             -BXI1(I,J)+BIX1(I,J))
140   CONTINUE
            C3=MXM2-MXM1+DLOG(DKXY2/DKXY1)
C
C
      DO 150 I=1,N
            SUM12=SUM12+B3(I)*X(I)
         DO 150 J=1,N
            SUM11=SUM11+X(I)*A3(I,J)*X(J)
150   CONTINUE
      RPX=0.5*(SUM11+SUM12+C3)
C
C
      DO 160 I=1,M
            SUM15=SUM15+B4(I)*Y(I)
         DO 160 J=1,M
            SUM14=SUM14+Y(I)*A4(I,J)*Y(J)
160   CONTINUE
      RY=0.5*(SUM14+SUM15+C4)
C
C
      VA=RPX+RY
C
C
      IF(VA.GT.T) THEN
           CLA=1
           NCLA1=NCLA1+1
      ELSE
           CLA=2
           NCLA2=NCLA2+1
      END IF
C
C
C                    ********************************
C    *************  CENTRALIZED DECISION RULE  ***********
C                    ********************************
C
200   DO 210 I=1,N
         XMX1(I)=X(I)-MX1(I)
         XMX2(I)=X(I)-MX2(I)
210   CONTINUE
C
C
      CALL VMULFF (BB1X,XMX1,M,N,1,M,N,BYT1,M,IER)
      CALL VMULFF (BB2X,XMX2,M,N,1,M,N,BYT2,M,IER)
C
C
      DO 220 I=1,M
         MBT1(I)=MY1(I)+BYT1(I)
         MBT2(I)=MY2(I)+BYT2(I)
220   CONTINUE
C
C
      CALL VMULFF (IKYX1,MBT1,M,M,1,M,M,KBT1,M,IER)
      CALL VMULFF (IKYX2,MBT2,M,M,1,M,M,KBT2,M,IER)
C
C
      CALL VMULFM (MBT1,IKYX1,M,1,M,M,M,MBK1,1,IER)
      CALL VMULFM (MBT2,IKYX2,M,1,M,M,M,MBK2,1,IER)
C
      CALL VMULFF (MBK1,MBT1,1,M,1,1,M,MT1,1,IER)
      CALL VMULFF (MBK2,MBT2,1,M,1,1,M,MT2,1,IER)
C
```

```
c
      DO 240 I=1,M
            B5(I)=KBT1(I)+MBK1(I)-(KBT2(I)+MBK2(I))
         DO 240 J=1,M
            A5(I,J)=IKYX2(I,J)-IKYX1(I,J)
240   CONTINUE
            C5=MT2-MT1+DLOG(DKYX2/DKYX1)
c
c
      DO 260 I=1,M
            SUM25=SUM25+B5(I)*Y(I)
         DO 260 J=1,M
            SUM24=SUM24+Y(I)*A5(I,J)*Y(J)
260   CONTINUE
      RBY=0.5*(SUM24+SUM25+C5)
c
c
      V=RX+RBY
c
c
      IF(V.GT.T) THEN
           CL=1
           NCL1=NCL1+1
      ELSE
           CL=2
           NCL2=NCL2+1
      END IF
c
      WRITE(7,*) VAL,VA,V
c
c     WRITE(7,298) V, T, CLASS, CLA, CL
c298  FORMAT (2X,E15.8,3X,F5.3,2X,3I7)
c
      GO TO 45
c
c
c299  RATEA1=100.*NCLAS1/L
c     RATEA2=100.*NCLAS2/L
c     RATEB1=100.*NCLA1/L
c     RATEB2=100.*NCLA2/L
c     RATEC1=100.*NCL1/L
c     RATEC2=100.*NCL2/L
c
c
c     WRITE(7,*) L,NCLAS1,NCLA1,NCL1
c     WRITE(7,*) RATEA1,RATEB1,RATEC1
c     WRITE(7,*) L,NCLAS2,NCLA2,NCL2
c     WRITE(7,*) RATEA2,RATEB2,RATEC2
299   STOP
      END
```

APPENDIX D

ANAL FORTRAN

```
c     This program counts the number of correct decisions
c     of three algorithms and calculates the correct
c     decision rates of them

      REAL*8 T,PRW1,PRW2,
     +       RATEA1,RATEA2,RATEB1,RATEB2,RATEC1,RATEC2,
     +       VAL(128),VA(128),V(128)
c
c
      INTEGER I,J,L,
     +        CLASS,CLA,CL,
     +        NCLAS1,NCLA1,NCL1,
     +        NCLAS2,NCLA2,NCL2
c
c
      L=128
c
c
      DO 10 I=1,L
         READ (2,*) VAL(I),VA(I),V(I)
10    CONTINUE
c
c
      PRW1=0.005
20    PRW2=1.-PRW1
c
      T=DLOG(PRW2/PRW1)
c
      NCLAS1=0
      NCLAS2=0
      NCLA1=0
      NCLA2=0
      NCL1=0
      NCL2=0
c
c
      DO 30 I=1,L
         IF (VAL(I).GT.T) THEN
            CLASS=1
            NCLAS1=NCLAS1+1
         ELSE
            CLASS=2
            NCLAS2=NCLAS2+1
         END IF
c
c
         IF (VA(I).GT.T) THEN
            CLA=1
            NCLA1=NCLA1+1
         ELSE
            CLA=2
            NCLA2=NCLA2+1
         END IF
c
c
         IF (V(I).GT.T) THEN
            CL=1
```

```
            NCL1=NCL1+1
         ELSE
            CL=2
            NCL2=NCL2+1
         END IF
C
C
30    CONTINUE
C
C
      RATEA1=100.*NCLAS1/L
      RATEA2=100.*NCLAS2/L
C
      RATEB1=100.*NCLA1/L
      RATEB2=100.*NCLA2/L
C
      RATEC1=100.*NCL1/L
      RATEC2=100.*NCL2/L
C
C     WRITE (7,*) CLASS,CLA,CL
C
C     WRITE (7,*) PRW1,PRW2,T,L
C
C     WRITE (7,*) NCLAS1,NCLA1,NCL1
C
      WRITE (7,*) RATEA1,RATEB1,RATEC1
C
C     WRITE (7,*) NCLAS2,NCLA2,NCL2
C
C     WRITE (7,*) RATEA2,RATEB2,RATEC2
C
C
      PRW1=PRW1+0.005
C
C
      IF (PRW1.GE.1.0) GO TO 299
C
      GO TO 20
C
C
299   STOP
      END
```

APPENDIX E

GRAPH4 DATA

These data files show the correct decision rates of 4 dimensional observation vectors. The first data is ANAL11 which represents case 1 and class 1 results. The capital letters "A" and "B" represent distributed decision rules A and B, and "C" means the centralized decision rule.
The varying prior probability Pr(w1) is given in the first column.

A N A L 1 1

Pr(wl)	NO. of correct decisions A	B	C	correct decision rates(%) A	B	C
0.050	52	52	52	40.625	40.625	40.625
0.100	62	64	63	48.438	50.000	49.219
0.150	70	70	72	54.688	54.688	56.250
0.200	77	75	79	60.156	58.594	61.719
0.250	81	82	83	63.281	64.063	64.844
0.300	88	91	89	68.750	71.094	69.531
0.350	95	95	95	74.219	74.219	74.219
0.400	99	98	100	77.344	76.563	78.125
0.450	105	105	104	82.031	82.031	81.250
0.500	110	110	109	85.938	85.938	85.156
0.550	110	110	112	85.938	85.938	87.500
0.600	113	114	114	88.281	89.063	89.063
0.650	115	115	116	89.844	89.844	90.625
0.700	119	119	119	92.969	92.969	92.969
0.750	120	120	120	93.750	93.750	93.750
0.800	122	121	121	95.313	94.531	94.531
0.850	123	123	123	96.094	96.094	96.094
0.900	124	125	124	96.875	97.656	96.875
0.950	125	125	126	97.656	97.656	98.438

A N A L 1 2

Pr(wl)	NO. of correct decisions A	B	C	correct decision rates(%) A	B	C
0.050	127	127	127	99.219	99.219	99.219
0.100	126	126	125	98.438	98.438	97.656
0.150	125	125	124	97.656	97.656	96.875
0.200	124	124	124	96.875	96.875	96.875
0.250	121	122	123	94.531	95.313	96.094
0.300	117	117	120	91.406	91.406	93.750
0.350	116	116	116	90.625	90.625	90.625
0.400	116	115	115	90.625	89.844	89.844
0.450	113	112	111	88.281	87.500	86.719
0.500	108	109	106	84.375	85.156	82.813
0.550	105	105	103	82.031	82.031	80.469
0.600	100	100	100	78.125	78.125	78.125
0.650	96	96	99	75.000	75.000	77.344
0.700	91	91	93	71.094	71.094	72.656
0.750	87	83	85	67.969	64.844	66.406
0.800	77	77	78	60.156	60.156	60.938
0.850	70	72	70	54.688	56.250	54.688
0.900	65	63	66	50.781	49.219	51.563
0.950	57	57	61	44.531	44.531	47.656

A N A L 2 1

Pr(wl)	NO. of correct decisions A	B	C	correct decision rates(%) A	B	C
0.050	86	88	82	67.188	68.750	64.063
0.100	104	100	91	81.250	78.125	71.094
0.150	108	104	102	84.375	81.250	79.688
0.200	109	109	107	85.156	85.156	83.594
0.250	110	112	111	85.938	87.500	86.719
0.300	110	114	111	85.938	89.063	86.719
0.350	113	114	113	88.281	89.063	88.281
0.400	115	120	116	89.844	93.750	90.625
0.450	117	120	118	91.406	93.750	92.188
0.500	119	120	118	92.969	93.750	92.188
0.550	119	120	119	92.969	93.750	92.969
0.600	121	122	121	94.531	95.313	94.531
0.650	123	123	123	96.094	96.094	96.094
0.700	124	123	123	96.875	96.094	96.094
0.750	124	123	124	96.875	96.094	96.875
0.800	125	124	125	97.656	96.875	97.656
0.850	125	124	127	97.656	96.875	99.219
0.900	126	127	127	98.438	99.219	99.219
0.950	127	127	128	99.219	99.219	100.000

A N A L 2 2

Pr(wl)	NO. of correct decisions A	B	C	correct decision rates(%) A	B	C
0.050	126	126	128	98.438	98.438	100.000
0.100	123	122	126	96.094	95.313	98.438
0.150	122	121	123	95.313	94.531	96.094
0.200	119	119	120	92.969	92.969	93.750
0.250	118	117	118	92.188	91.406	92.188
0.300	116	117	117	90.625	91.406	91.406
0.350	115	115	117	89.844	89.844	91.406
0.400	114	113	115	89.063	88.281	89.844
0.450	112	113	114	87.500	88.281	89.063
0.500	109	110	114	85.156	85.938	89.063
0.550	109	109	110	85.156	85.156	85.938
0.600	107	108	107	83.594	84.375	83.594
0.650	107	108	104	83.594	84.375	81.250
0.700	102	103	104	79.688	80.469	81.250
0.750	102	103	101	79.688	80.469	78.906
0.800	101	98	96	78.906	76.563	75.000
0.850	97	93	91	75.781	72.656	71.094
0.900	88	91	83	68.750	71.094	64.844
0.950	79	80	70	61.719	62.500	54.688

A N A L 3 1

Pr(wl)	NO. of correct decisions A	B	C	correct decision rates(%) A	B	C
0.050	58	54	59	45.313	42.188	46.094
0.100	69	67	68	53.906	52.344	53.125
0.150	74	74	76	57.813	57.813	59.375
0.200	77	76	77	60.156	59.375	60.156
0.250	83	83	85	64.844	64.844	66.406
0.300	84	86	89	65.625	67.188	69.531
0.350	86	87	95	67.188	67.969	74.219
0.400	93	94	100	72.656	73.438	78.125
0.450	99	102	106	77.344	79.688	82.813
0.500	104	107	109	81.250	83.594	85.156
0.550	109	109	112	85.156	85.156	87.500
0.600	112	113	115	87.500	88.281	89.844
0.650	114	114	116	89.063	89.063	90.625
0.700	117	116	117	91.406	90.625	91.406
0.750	118	117	120	92.188	91.406	93.750
0.800	120	120	120	93.750	93.750	93.750
0.850	122	121	122	95.313	94.531	95.313
0.900	124	124	124	96.875	96.875	96.875
0.950	126	126	126	98.438	98.438	98.438

A N A L 3 2

Pr(wl)	NO. of correct decisions A	B	C	correct decision rates(%) A	B	C
0.050	126	126	127	98.438	98.438	99.219
0.100	125	125	125	97.656	97.656	97.656
0.150	124	124	124	96.875	96.875	96.875
0.200	122	122	124	95.313	95.313	96.875
0.250	120	120	120	93.750	93.750	93.750
0.300	117	117	117	91.406	91.406	91.406
0.350	116	116	116	90.625	90.625	90.625
0.400	115	115	114	89.844	89.844	89.063
0.450	115	114	114	89.844	89.063	89.063
0.500	110	111	110	85.938	86.719	85.938
0.550	103	103	103	80.469	80.469	80.469
0.600	99	101	101	77.344	78.906	78.906
0.650	97	96	98	75.781	75.000	76.563
0.700	90	90	93	70.313	70.313	72.656
0.750	88	89	88	68.750	69.531	68.750
0.800	84	86	82	65.625	67.188	64.063
0.850	79	80	75	61.719	62.500	58.594
0.900	71	68	66	55.469	53.125	51.563
0.950	57	56	58	44.531	43.750	45.313

ANAL41

Pr(wl)	NO. of correct decisions A	B	C	correct decision rates(%) A	B	C
0.050	3	4	1	2.344	3.125	0.781
0.100	9	8	2	7.031	6.250	1.563
0.150	16	14	3	12.500	10.938	2.344
0.200	21	23	9	16.406	17.969	7.031
0.250	30	34	14	23.438	26.563	10.938
0.300	43	44	22	33.594	34.375	17.188
0.350	55	55	34	42.969	42.969	26.563
0.400	70	71	48	54.688	55.469	37.500
0.450	89	90	61	69.531	70.313	47.656
0.500	110	112	73	85.938	87.500	57.031
0.550	123	123	86	96.094	96.094	67.188
0.600	127	127	109	99.219	99.219	85.156
0.650	128	128	118	100.000	100.000	92.188
0.700	128	128	125	100.000	100.000	97.656
0.750	128	128	126	100.000	100.000	98.438
0.800	128	128	128	100.000	100.000	100.000
0.850	128	128	128	100.000	100.000	100.000
0.900	128	128	128	100.000	100.000	100.000
0.950	128	128	128	100.000	100.000	100.000

ANAL42

Pr(wl)	NO. of correct decisions A	B	C	correct decision rates(%) A	B	C
0.050	128	128	128	100.000	100.000	100.000
0.100	127	125	128	99.219	97.656	100.000
0.150	125	125	128	97.656	97.656	100.000
0.200	122	122	128	95.313	95.313	100.000
0.250	114	116	127	89.063	90.625	99.219
0.300	109	106	125	85.156	82.813	97.656
0.350	96	95	122	75.000	74.219	95.313
0.400	79	78	114	61.719	60.938	89.063
0.450	49	47	98	38.281	36.719	76.563
0.500	25	22	77	19.531	17.188	60.156
0.550	10	11	62	7.813	8.594	48.438
0.600	3	3	39	2.344	2.344	30.469
0.650	1	0	23	0.781	0.000	17.969
0.700	0	0	13	0.000	0.000	10.156
0.750	0	0	10	0.000	0.000	7.813
0.800	0	0	4	0.000	0.000	3.125
0.850	0	0	3	0.000	0.000	2.344
0.900	0	0	1	0.000	0.000	0.781
0.950	0	0	1	0.000	0.000	0.781

APPENDIX F

GRAPH32 DATA

These data files show the correct decision rates for 32 dimensional observation vectors. The first data is ANAL11 which represents case 1 and class 1 results. The capital letters "A" and "B" represent distributed decision rules A and B, and "C" means the centralized decision rule.
The varying prior probability Pr(w1) is given in the first column.

A N A L 1 1

Pr(wl)	NO. of correct decisions A	B	C	correct decision rates(%) A	B	C
0.050	128	128	128	100.000	100.000	100.000
0.100	128	128	128	100.000	100.000	100.000
0.150	128	128	128	100.000	100.000	100.000
0.200	128	128	128	100.000	100.000	100.000
0.250	128	128	128	100.000	100.000	100.000
0.300	128	128	128	100.000	100.000	100.000
0.350	128	128	128	100.000	100.000	100.000
0.400	128	128	128	100.000	100.000	100.000
0.450	128	128	128	100.000	100.000	100.000
0.500	128	128	128	100.000	100.000	100.000
0.550	128	128	128	100.000	100.000	100.000
0.600	128	128	128	100.000	100.000	100.000
0.650	128	128	128	100.000	100.000	100.000
0.700	128	128	128	100.000	100.000	100.000
0.750	128	128	128	100.000	100.000	100.000
0.800	128	128	128	100.000	100.000	100.000
0.850	128	128	128	100.000	100.000	100.000
0.900	128	128	128	100.000	100.000	100.000
0.950	128	128	128	100.000	100.000	100.000

A N A L 1 2

Pr(wl)	NO. of correct decisions A	B	C	correct decision rates(%) A	B	C
0.050	128	128	128	100.000	100.000	100.000
0.100	128	128	128	100.000	100.000	100.000
0.150	128	128	128	100.000	100.000	100.000
0.200	128	128	128	100.000	100.000	100.000
0.250	128	128	128	100.000	100.000	100.000
0.300	128	128	128	100.000	100.000	100.000
0.350	128	128	128	100.000	100.000	100.000
0.400	128	128	128	100.000	100.000	100.000
0.450	128	128	128	100.000	100.000	100.000
0.500	128	128	128	100.000	100.000	100.000
0.550	128	128	128	100.000	100.000	100.000
0.600	128	128	128	100.000	100.000	100.000
0.650	128	128	128	100.000	100.000	100.000
0.700	128	128	128	100.000	100.000	100.000
0.750	128	128	128	100.000	100.000	100.000
0.800	128	128	128	100.000	100.000	100.000
0.850	128	128	128	100.000	100.000	100.000
0.900	128	128	128	100.000	100.000	100.000
0.950	128	128	128	100.000	100.000	100.000

A N A L 2 1

Pr(wl)	NO. of correct decisions A	B	C	correct decision rates(%) A	B	C
0.050	128	128	128	100.000	100.000	100.000
0.100	128	128	128	100.000	100.000	100.000
0.150	128	128	128	100.000	100.000	100.000
0.200	128	128	128	100.000	100.000	100.000
0.250	128	128	128	100.000	100.000	100.000
0.300	128	128	128	100.000	100.000	100.000
0.350	128	128	128	100.000	100.000	100.000
0.400	128	128	128	100.000	100.000	100.000
0.450	128	128	128	100.000	100.000	100.000
0.500	128	128	128	100.000	100.000	100.000
0.550	128	128	128	100.000	100.000	100.000
0.600	128	128	128	100.000	100.000	100.000
0.650	128	128	128	100.000	100.000	100.000
0.700	128	128	128	100.000	100.000	100.000
0.750	128	128	128	100.000	100.000	100.000
0.800	128	128	128	100.000	100.000	100.000
0.850	128	128	128	100.000	100.000	100.000
0.900	128	128	128	100.000	100.000	100.000
0.950	128	128	128	100.000	100.000	100.000

A N A L 2 2

Pr(wl)	NO. of correct decisions A	B	C	correct decision rates(%) A	B	C
0.050	128	128	128	100.000	100.000	100.000
0.100	128	128	128	100.000	100.000	100.000
0.150	128	128	128	100.000	100.000	100.000
0.200	128	128	128	100.000	100.000	100.000
0.250	128	128	128	100.000	100.000	100.000
0.300	128	128	128	100.000	100.000	100.000
0.350	128	128	128	100.000	100.000	100.000
0.400	128	128	128	100.000	100.000	100.000
0.450	128	128	128	100.000	100.000	100.000
0.500	128	128	128	100.000	100.000	100.000
0.550	0	128	128	0.000	100.000	100.000
0.600	0	128	128	0.000	100.000	100.000
0.650	0	128	128	0.000	100.000	100.000
0.700	0	128	128	0.000	100.000	100.000
0.750	0	128	128	0.000	100.000	100.000
0.800	0	128	128	0.000	100.000	100.000
0.850	0	128	128	0.000	100.000	100.000
0.900	0	128	128	0.000	100.000	100.000
0.950	0	128	128	0.000	100.000	100.000

A N A L 3 1

	NO. of correct decisions			correct decision rates(%)		
Pr(wl)	A	B	C	A	B	C
0.050	128	128	128	100.000	100.000	100.000
0.100	128	128	128	100.000	100.000	100.000
0.150	128	128	128	100.000	100.000	100.000
0.200	128	128	128	100.000	100.000	100.000
0.250	128	128	128	100.000	100.000	100.000
0.300	128	128	128	100.000	100.000	100.000
0.350	128	128	128	100.000	100.000	100.000
0.400	128	128	128	100.000	100.000	100.000
0.450	128	128	128	100.000	100.000	100.000
0.500	128	128	128	100.000	100.000	100.000
0.550	128	128	128	100.000	100.000	100.000
0.600	128	128	128	100.000	100.000	100.000
0.650	128	128	128	100.000	100.000	100.000
0.700	128	128	128	100.000	100.000	100.000
0.750	128	128	128	100.000	100.000	100.000
0.800	128	128	128	100.000	100.000	100.000
0.850	128	128	128	100.000	100.000	100.000
0.900	128	128	128	100.000	100.000	100.000
0.950	128	128	128	100.000	100.000	100.000

A N A L 3 2

	NO. of correct decisions			correct decision rates(%)		
Pr(wl)	A	B	C	A	B	C
0.050	128	128	128	100.000	100.000	100.000
0.100	128	128	128	100.000	100.000	100.000
0.150	128	128	128	100.000	100.000	100.000
0.200	128	128	128	100.000	100.000	100.000
0.250	128	128	128	100.000	100.000	100.000
0.300	128	128	128	100.000	100.000	100.000
0.350	128	128	128	100.000	100.000	100.000
0.400	128	128	128	100.000	100.000	100.000
0.450	128	128	128	100.000	100.000	100.000
0.500	127	127	120	99.219	99.219	93.750
0.550	127	127	120	99.219	99.219	93.750
0.600	127	127	120	99.219	99.219	93.750
0.650	127	127	120	99.219	99.219	93.750
0.700	127	127	120	99.219	99.219	93.750
0.750	127	127	120	99.219	99.219	93.750
0.800	127	127	120	99.219	99.219	93.750
0.850	127	127	120	99.219	99.219	93.750
0.900	127	127	120	99.219	99.219	93.750
0.950	127	127	120	99.219	99.219	93.750

A N A L 4 1

Pr(wl)	NO. of correct decisions A	B	C	correct decision rates(%) A	B	C
0.050	103	108	69	80.469	84.375	53.906
0.100	116	119	83	90.625	92.969	64.844
0.150	121	123	87	94.531	96.094	67.969
0.200	125	124	92	97.656	96.875	71.875
0.250	125	126	96	97.656	98.438	75.000
0.300	125	126	101	97.656	98.438	78.906
0.350	127	126	106	99.219	98.438	82.813
0.400	127	126	108	99.219	98.438	84.375
0.450	127	128	111	99.219	100.000	86.719
0.500	128	128	113	100.000	100.000	88.281
0.550	128	128	113	100.000	100.000	88.281
0.600	128	128	114	100.000	100.000	89.063
0.650	128	128	117	100.000	100.000	91.406
0.700	128	128	120	100.000	100.000	93.750
0.750	128	128	122	100.000	100.000	95.313
0.800	128	128	124	100.000	100.000	96.875
0.850	128	128	125	100.000	100.000	97.656
0.900	128	128	127	100.000	100.000	99.219
0.950	128	128	127	100.000	100.000	99.219

A N A L 4 2

Pr(wl)	NO. of correct decisions A	B	C	correct decision rates(%) A	B	C
0.050	101	87	127	78.906	67.969	99.219
0.100	77	69	125	60.156	53.906	97.656
0.150	63	56	124	49.219	43.750	96.875
0.200	55	50	123	42.969	39.063	96.094
0.250	46	43	122	35.938	33.594	95.313
0.300	36	33	122	28.125	25.781	95.313
0.350	29	26	121	22.656	20.313	94.531
0.400	25	21	120	19.531	16.406	93.750
0.450	21	11	119	16.406	8.594	92.969
0.500	17	8	117	13.281	6.250	91.406
0.550	12	8	116	9.375	6.250	90.625
0.600	7	7	114	5.469	5.469	89.063
0.650	5	7	112	3.906	5.469	87.500
0.700	4	3	112	3.125	2.344	87.500
0.750	3	3	105	2.344	2.344	82.031
0.800	2	1	100	1.563	0.781	78.125
0.850	1	0	93	0.781	0.000	72.656
0.900	0	0	87	0.000	0.000	67.969
0.950	0	0	71	0.000	0.000	55.469

LIST OF REFERENCES

1. Schon, M. A., Development of a Testbed for Multisensor Distributed Decision Algorithms, M.S. Thesis, Naval Postgraduate School, Monterey, California, December 1985.

2. Helstrom, C. W., Probability and Stochastic Processes for Engineers, Macmillan Publishing Company, New York 1982.

3. Klinefelter, S.G., Implementation of a Real-Time, Distributed Operating System for a Multiple Computer System, M.S. Thesis, Naval Postgraduate School, Monterey, California, June 1982.

4. Van Trees, H.L., Detection Estimation and Modulation Theory, Part I, John Wiley & Sons, New York, 1968.

5. International Mathematical & Statistical Libraries, Inc., IMSL Library, IMSL, Inc. , November 1984.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5100	2
2.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
3.	Department Chairman, code 62 Dept. of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5100	1
4.	Prof. Charles W. Therrien, Code 62 Ti Dept. of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5100	3
5.	Prof. Uno R. Kodres, Code 52 Kr Dept. of Computer Science Naval Postgraduate School Monterey, California 93943-5100	2
6.	Prof. M. L. Cotton, Code 62 Cc Dept. of Electrical and Computer Science Naval Postgraduate School Monterey, California 93943-5100	1
7.	Prof. R. Panholzer, Code 62 Pz Dept. of Electrical and Computer Science Naval Postgraduate School Monterey, California 93943-5100	1
8.	Maj. Sung-Chu Hahn PO #17 Daebang-dong Dongjakgu Seoul 151-01 Republic of Korea	7
9.	Capt. Mark A. Schon 1801 Artillery Ridge Road Fredericksburg, Virginia 22401	1
10.	Professor P. Moose, Code 62 Me Dept. of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5100	1
11.	Mr. D. Cowan NWC China Lake, Code 31507 China Lake, California 93555	1
12.	Dr. P. Krueger Air 320H Naval Air Systems Command Washington, D. C. 20361	1